UNIVERSITY OF ALBERTA

# Continuous And Discrete Wavelet Transforms And Finite Difference Methods For Parabolic Equations

*BY*

© Shuzhan Xu

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
MATHEMATICS

DEPARTMENT OF MATHEMATICAL SCIENCES

EDMONTON, ALBERTA
Spring, 1995

ISBN 0-612-01779-6

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: **Shuzhan Xu**

TITLE OF THESIS: **Continuous And Discrete Wavelet Transforms And Finite Difference Methods For Parabolic Equations**

DEGREE FOR WHICH THESIS WAS PRESENTED: **Doctor of Philosophy**

YEAR THIS DEGREE GRANTED: **1995**

PERMISSION IS HEREBY GRANTED TO THE UNIVERSITY OF ALBERTA LIBRARY TO REPRODUCE SINGLE COPIES OF THIS THESIS AND TO LEND OR SELL SUCH COPIES FOR PRIVATE, SCHOLARLY OR SCIENTIFIC RESEARCH PURPOSES ONLY.

THE AUTHOR RESERVES OTHER PUBLICATION RIGHTS, AND NEITHER THE THESIS NOR EXTENSIVE EXTRACTS FROM IT MAY BE PRINTED OR OTHERWISE REPRODUCED WITHOUT THE AUTHOR'S WRITTEN PERMISSION.

Xu Shuzhan

Signature

Dept. of Mathematial Sciences

Permanent Address

Univ. of Alberta
Edmonton, Canada
T6G 2G1

Date: Jan 26, 95

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

THE UNDERSIGNED CERTIFY THAT THEY HAVE READ, AND RECOMMEND TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH FOR ACCEPTANCE, A THESIS ENTITLED **CONTINUOUS AND DISCRETE WAVELET TRANSFORMS AND FINITE DIFFERENCE METHODS FOR PARABOLIC EQUATIONS** SUBMITTED BY **SHUZHAN XU** IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF **DOCTOR OF PHILOSOPHY** IN MATHEMATICS.

SD Riemenschneider

S. D. Riemenschneider

Y. Lin

R. Q. Jia

Z J Koles

Z. J. Koles

S. Shen

Herman Brunner

H. Brunner

Date: 25 January 95

*To my wife and my parents*

# ABSTRACT

This thesis deals with two numerical schemes: wavelet transforms and finite difference approximation schemes. As a particular application of the discrete wavelet transform, image compression is also discussed.

Wavelet analysis gives us a way to decompose functions by using scaling and shift of a single function. In real applications, efficient algorithms to compute wavelet transforms must be studied. We first present a new algorithm to compute continuous wavelet transform at dyadic scales by applying quasi-interpolation and subdivision techniques. Our algorithm is well connected to the standard *algorithme à trous* and can be considered as its extension. For discrete wavelet transform, we mainly study wavelet transform coding and its application to image compression. Different compression schemes and their behavior are studied with support of experiments and examples.

For finite difference approximation, we first transform a inverse problem with an unknown source parameter into an equivalent non-local parabolic equation. Finite difference scheme is formulated for the new problem and the unknown source parameter can be calculated through the inverse transformation via numerical differentiation. We then study finite difference approximations to the solutions of some non-local parabolic equations subject to certain kernel condition. The proposed finite difference procedures preserve many properties of the solution. Both of these two equations are analyzed by using the discrete version (modified) of the maximum principle for parabolic finite difference schemes. Numerical computations are also presented to support our theoretical analysis.

# ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Professor Sherman Riemenschneider, for his excellent guidance, great encouragement and generous support through my entire Ph.D program.

It is my great pleasure to acknowledge my gratitude to my teacher and friend Professor Yanping Lin for his valuable guidance in the area of numerical analysis. I would like to take this opportunity to express my deep appreciation for his encouragement and friendship. Special thanks also goes to Professor Rong Qing Jia for his concern and many valuable suggestions concerning the wavelet analysis.

I am very grateful to Professors Hermann Brunner, Samuel Shen and Zolly Koles for their interest in my research and their valuable suggestions. Also I thank Jeroen Bielen for allowing me to use his picture for image compression and Mark Astridge for helping me to prepare some images.

I would like to thank Department of Mathematics, University of Alberta for its support during my program of study.

I am also greatly indebted to many of my friends in Edmonton who have supported me so much in various ways. Their friendship and concern made my venture in Edmonton became a never forgetable chapter of my whole life.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## §1.0. Preamble

In this thesis, we will study two numerical schemes: wavelet transforms and finite difference approximation. As a particular application of the wavelet transform, image compression is also discussed. As outlined in the following sections, we will study a new algorithm to compute the continuous wavelet transform in Chapter 2 and in Chapter 3 we will discuss wavelet transform coding and image compression. In Chapter 4 and Chapter 5, we will study finite difference solutions of two parabolic equations which are both nonlocal and can be analyzed by the discrete maximum principle. In what follows, we may refer to a function as a signal or the other way around. For $f, g \in L_2(R^n)$,

$$< f, g >= \int_{R^n} f(t)g(t)dt.$$

## §1.1. Compute Continuous Wavelet Transform By Subdivision

The wavelet transform gives us a way to decompose functions by using scaling and shift of a single function $\psi(t)$ called a wavelet. The continuous wavelet transform of a function $f$ defined on $R^n$ corresponding to the analyzing wavelet $\psi(t)$ is defined as:

$$\mathcal{T}f(b,a) = \int_{R^n} f(t)\psi_{b,a}(t)dt$$

where $b \in R^n, a \in R, a > 0$, and $\psi_{b,a}(t) = a^{-\frac{n}{2}}\psi(\frac{t-b}{a})$. Only very mild assumptions are needed for the analyzing wavelet, therefore we have the freedom to choose the right wavelet. This time-scale approach turned out to be very efficient [4][5][6][8][12][13]. Wavelet transforms have some nice properties; notably,

- Localization property: Normally we require $\psi(t)$ to be either compactly supported or exponential decay in both time and frequency domain and thus get local analysis.
- Reproducing property: If $K(a_0, a, b_0, b) = <\psi_{b,a}, \psi_{b_0,a_0}>$, then

  $$\mathcal{T}f(b_0, a_0) = \int_0^{+\infty}\int_{R^n} \mathcal{T}f(b,a)K(a_0, a, b_0, b)db\,da$$

  which shows the redundancy of the information contained in the transform and thus gives us the ability to cross reference [6][8][12][13].

In real applications [6][8][12][13], efficient algorithms are needed to compute wavelet transform. For the continuous wavelet transform, Holschneider, Kronland-Martinet, Morlet and Tchamitchian [20] proposed the efficient *algorithme à trous* which is based on a *hole filling process.* The hole filling process is limited in the sense that it uses spline interpolation that is restricted to piecewise constant and piecewise linear splines. In search of an efficient algorithm with higher order accuracy, we extend the *algorithme à trous* by just approximating the analyzing wavelet by a linear combination of refinable functions; approximating by quasi-interpolation in the spline case. The refinability permits subdivision schemes to be applied in the computation which lead to a very efficient new algorithm. The algorithm has an almost identical implementation as the *algorithme à trous* but the starting point is different. Our algorithm coincides with the *algorithme à trous* in the one dimensional case if the initial approximation is given by piecewise constant or piecewise linear spline interpolation. This coincidence reveals the connection between subdivision and hole filling, which are two basic approaches to render given data and provide approximation. The algorithm is an extension of the *algorithme à trous* for it provides higher order approximation and easily generalizes to higher dimensions. We will discuss these algorithms in Chapter 2.

## §1.2. Image Compression By FWT

Mathematically, it is equivalent to study a function or the coefficients in the series expansion of the function under certain bases. Viewing series expansion as a transform, we gain the transform coding scheme: code the coefficients of the series expansion. For given signals, with the right basis the series expansions could be very sparse and most of the coefficients nearly equal to zero. By ignoring these small coefficients and keeping only the large coefficients, we may still get a good approximation of the original signals. This gives what is called a lossy compression scheme: there is some loss of data. Orthogonal bases are naturally better suited to this scheme and normally bases which possess fast transforms are applied since efficient computation is needed. In Chapter 3, we will study this compression scheme in a general setting and discuss several aspects. All our discussion is limited to the one dimensional case since we use tensor product approach to deal with image compression. Experiments of image compression are presented to support our discussion.

The efficiency of this scheme obviously depends on both signal and basis, more precisely it depends on the sparsity of the series expansion. Our experiments are mainly done by applying orthogonal wavelets: specially constructed $\psi(x)$ such that $\{\psi(2^i x - j)\}_{i,j\in Z}$ form an orthogonal basis of $L_2(R)$. Developed by Mallat [13], the FWT (fast wavelet transform) is an efficient algorithm to compute the coefficients of wavelet series or continuous wavelet transform at dyadic values. FWT is also known as Mallat's algorithm, or the discrete wavelet transform, or the orthogonal wavelet transform. With locality and scaling/shift nature, wavelets are extremely efficient to match a change in the signal [4][5][6][8][12]. In other words, wavelets can mimic the shape of the signal better than ordinary bases and the wavelet series is often very sparse. As our example shows, orthogonal wavelets are especially suited for compression.

## §1.3. A Finite Difference Scheme For An Inverse Heat Equation

In Chapter 4, we will study a finite difference method for approximating the unknown source parameter $p = p(t)$ and $u = u(x, y, t)$ of the following inverse problem. Find $u = u(x, y, t)$ and $p = p(t)$ which satisfy

$$\begin{aligned} &u_t = \Delta u + p(t)u + f(x, y, t) \ \text{ in } \ Q_T, \\ &u(x, y, 0) = \phi(x, y), \qquad (x, y) \in \Omega, \\ &u(x, y, t) = g(x, y, t), \qquad \text{on } \partial\Omega \times [0, T], \end{aligned}$$

subject to the additional constraints

$$u(x^*, y^*, t) = E(t) \qquad \text{for fixed} \quad (x^*, y^*) \in \Omega, \ \ 0 \le t \le T,$$

where $Q_T = \Omega \times (0, T]$, $T > 0$, $\Omega = (0, 1) \times (0, 1)$, $f, \phi, g$ and $E \neq 0$ are known functions, and $(x^*, y^*)$ is a fixed prescribed interior point in $\Omega$. As usual, the boundary of $\Omega$ is denoted by $\partial\Omega$. In real applications, if $u$ represents temperature then the problem can be viewed as a control problem of finding the control $p = p(t)$ such that the internal constraint is satisfied. The problem above and other similar inverse problems of identifying some unknown source parameter have been studied by several authors recently with different approaches [18][19][21][22]. We will study the backward

Euler finite difference scheme, and this scheme will be shown to be stable in the maximum norm by using the discrete version (modified) of the maximum principle for parabolic finite difference schemes.

Notice that this problem cannot be easily solved by finite difference scheme directly. Therefore we define the following transformation:

$$v(x,y,t) = u(x,y,t)\exp\{-\int_0^t p(s)ds\}, \qquad r(t) = \exp\{-\int_0^t p(s)ds\}.$$

Then

$$u(x,y,t) = \frac{v(x,y,t)}{r(t)}, \qquad p(t) = \frac{-r'(t)}{r(t)},$$

and the problem is transformed into the following parabolic equation

$$\begin{aligned} &v_t = \Delta v + r(t)f(x,y,t) \text{ in } Q_T, \\ &v(x,y,0) = \phi(x,y), && (x,y) \in \Omega, \\ &v(x,y,t) = r(t)g(x,y,t), && \text{on } \partial\Omega \times [0,T], \end{aligned}$$

subject to

$$r(t) = \frac{v(x^*,y^*,t)}{E(t)} \quad \text{for fixed } (x^*,y^*) \in \Omega, \ 0 \le t \le T.$$

This problem is equivalent to the original one provided that the data is smooth enough. Therefore a finite difference scheme is formulated for this problem, and the scheme will be shown to be stable in the maximum norm. Once $v$ is known numerically the unknown $(u,p)$ can be calculated through the inverse transformation via numerical differentiation. By controlling the step size in the numerical differentiation, we can demonstrate the convergence of the approximation to $u$ and $p$. Several aspects, particularly the stability, of our numerical procedure are discussed. Also we present some numerical computations for several examples to support our theoretical analysis. Some extensions are also addressed.

## §1.4. A Class of Non-Local Parabolic Equations

In the final Chapter, we will study finite difference approximations to the solution of the following non-local parabolic equations:

$$
\begin{aligned}
&u_t - \Delta u = 0 \qquad \text{in } Q_T, \\
&u(x,y,0) = \phi(x,y), \qquad (x,y) \in \Omega, \\
&u(x,y,t) = \int_\Omega K(x,y,\xi,\eta)u(\xi,\eta,t)d\xi d\eta \qquad \text{on } \partial\Omega \times [0,T),
\end{aligned}
$$

where $Q_T = \Omega \times (0,T)$, $T > 0$, $\Omega = (0,1) \times (0.1)$, $\phi(x,y) \not\equiv 0$ and $K(x,y,\xi,\eta)$ are known functions. In addition, it is assumed that for some constant $0 < \rho < 1$ the kernel $K(x,y,\xi,\eta)$ satisfies

$$\int_\Omega |K(x,y,\xi,\eta)|d\xi d\eta \leq \rho < 1, \qquad \forall (x,y) \in \partial\Omega.$$

Equations similar to our problem here arise from some real problems, the determination of the unknown source parameter and other related problems [3][7]. And they have been studied in different contexts. We will study finite difference approximations for the original problem. The proposed finite difference procedures preserve monotonicity, the maximum principle and the exponential decay (if the kernel is non-negative) of the solution for the original equation; therefore, they are considered as good numerical approximations.

With the help of numerical integration, we propose both fully implicit and semi-implicit schemes. The two schemes differ in the treatment of the condition

$$\int_\Omega |K(x,y,\xi,\eta)|d\xi d\eta \leq \rho < 1, \qquad \forall (x,y) \in \partial\Omega$$

which is necessary in order to obtain numerical solutions that preserve as many properties of the solution as possible. Numerically, the fully-implicit scheme requires a full-matrix system to be solved at each time level due to the boundary integration. With small step sizes, the matrix will be diagonally-dominant and the corresponding linear system can be solved by any standard method. The semi-implicit scheme is easy to implement numerically since only a pent-diagonal matrix system needs to be solved at each time level. Therefore, it is a very economical and fast algorithm. Both the fully implicit and semi-implicit schemes result in an truncation error $O(h^2 + \tau)$. We will prove that under the condition

$$\int_\Omega |K(x, y, \xi, \eta)| d\xi d\eta \leq \rho < 1, \qquad \forall (x, y) \in \partial\Omega,$$

the numerical solutions of the two schemes are unconditionally stable. On the other hand, if this condition is not satisfied, both of these two schemes may lose unconditional stability as demonstrated by our numerical computations. The conditional stability of general cases is also discussed.

## REFERENCES

1. C. de Boor, A practical guide to splines, Springer-Verlag, New York, 1984.

2. C. de Boor, K. Höllig and S. Riemenschneider, Box splines, Springer-Verlag, New York, 1993.

3. J. R. Cannon, The one-dimensional heat equation, Encyclopedia of Mathematics and Its Applications, Vol 23, Addison-Wesley, Reading, Massachusetts, 1984.

4. C. Chui, An introduction to wavelets, Academic Press, New York, 1992.

5. R. Coifman, Adapted multiresolution analysis, computation, signal processing and operator theory, ICM 90, Kyoto, Japan, Springer-Verlag, New York, 1990.

6. J. M. Combes, A. Grossman and P. Tchamitchian eds., Wavelets: Time-frequency methods and phase space, Springer-Verlag, Berlin, 1990.

7. W. A. Day, Heat conduction within linear thermoelasticity, Springer-Verlag, New York, 1985.

8. I. Daubechies, Ten lectures on wavelets, CBMS 61 SIAM, Philadelphia, 1992.

9. G. E. Forsythe and W. R. Wason, Finite-difference methods for partial differential equations, John Wiley and Sons, New York, 1960.

10. A. Friedman, Partial differential equations of parabolic type, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.

11. H. S. Malvar, Signal processing with lapped transforms, Artec House, Norwood, MA, 1992.

12. Y. Meyer, Ondelettes, Hermann, New York, 1988.

13. Y. Meyer, Wavelets, algorithms & applications, SIAM, Philadelphia, PA, 1993.

14. A. Oppenheim and R. W. Schafer, Discrete-time signal processing, Prentice Hall, Englewood Cliffs, New Jersey, 1989.

15. W. Pratt, Digital image processing, John Wiley and Sons, New York, 1992.

16. I. Schoenberg, Cardinal spline interpolation, SIAM, Philadelphia, PA, 1973.

17. A. Zygmund, Trigonometric series, Cambridge University Press, Cambridge, 1959.

## ADDITIONAL REFERENCES

18. J. R. Cannon and Y. Lin, An inverse problem of finding a parameter in a semilinear heat equation, J. Math. Anal. Appl., 2(145), (1990), 470-484.

19. J. R. Cannon, Y. Lin and S. Wang, Determination of source parameter in parabolic equations, MECCANICA, 27(1992), 85-94.

20. M. Holschneider, R. Kronland-Martinet, J. Morlet and P. Tchamitchian, A realtime algorithm for signal analysis with the help of wavelet transform, Wavelets: Time-frequency methods and phase space, J. M. Combes et al. eds., Springer-Verlag, Berlin, 1990, 286-297.

21. Y. Lin, An inverse problem for a class of quasilinear parabolic equations, SIAM J. Math. Anal., 22(1991), 146-156.

22. A. I. Prilepko and V. V. Solo'ev, Solvability of the inverse boundary value problem of finding a coefficient of a lower order term in a parabolic equation, Diff. Eqs., 1(23), 1987, 136-143.

# Chapter 2

# Compute Continuous Wavelet Transform By Quasi-Interpolation and Subdivision Scheme

## §2.0. Introduction

The idea of wavelet analysis is to decompose functions by using scaling and shift of a single function $\psi(t)$ called a wavelet. The **continuous wavelet transform(CWT)**, or **integral wavelet transform(IWT)**, of a function $f$ defined on $R^n$ corresponding to an analyzing wavelet $\psi(t)$ is defined as:

$$\mathcal{T}f(b,a) = \int_{R^n} f(t)\psi_{b,a}(t)dt$$

where $b \in R^n, a \in R, a > 0$, and $\psi_{b,a}(t) = a^{-\frac{n}{2}}\psi(\frac{t-b}{a})$. Usually compactly supported or exponentially decaying wavelets are used in order to get local analysis. The fundamental result governing the continuous wavelet transform is the following conclusion by Grossman and Morlet [11][16]:

**Theorem 0.1.** *If $f(t) \in L_2(R^n)$, then the following equations*

$$\int_{R^n} |f|^2 dt = \frac{1}{c(\psi)}\int_{R^n}\int_0^\infty |\mathcal{T}f(b,a)|^2 \frac{dadb}{a^2},$$

$$f = \frac{1}{c(\psi)}\int_{R^n}\int_0^\infty \mathcal{T}f(b,a)\psi_{b,a}\frac{dadb}{a^2}$$

*hold if and only if*

$$c(\psi) = (2\pi)^n \int_{R^n} \frac{|\hat{\psi}(\omega)|^2}{|\omega|^n} d\omega < \infty.$$

Note that only very mild assumptions are imposed on the analyzing wavelet, therefore we have the freedom to choose the right wavelet for a particular application. By using very smooth wavelets, the continuous wavelet transform provides a very powerful analysis tool which can be applied to several different areas [11][22][23][24].

Numerically, the CWT can be computed by FFT based on the representation in the frequency domain, but this is too expensive computationally [14]. Yet so many real applications depend on the computation of CWT that an efficient algorithm to compute it must be studied.

The standard fast algorithm to compute CWT is the *algorithme à trous* developed by Holschneider, Kronland-Martinet, Morlet and Tchamitchian [19] which is based on a *hole filling* process. When the hole filling process is viewed as a spline interpolation process, the algorithm cannot accommodate spline interpolation of order higher than piecewise constant and piecewise linear [19]. Moreover, this algorithm was only applied to the one dimensional case.

We introduce a new algorithm to compute the CWT. As opposed to hole filling, we use approximation of the analyzing wavelet by refinable functions. Because of refinability, subdivision schemes can be applied in the computation and this leads to efficient algorithms. We present a clear and general theoretical background for the computation of CWT through this approach. In particular, if cardinal B-spline and interpolation are applied with order no bigger than 2, our algorithm coincides with the *algorithme à trous*.

This chapter is organized as follows: in §2.1, we discuss the problems around the computation of continuous wavelet transform and present the *algorithme à trous*. The new algorithm will be presented in §2.2. In §2.3, we discuss the special case of cardinal B-splines and quasi-interpolations and show how our approach connects with the *algorithme à trous*. In §2.4, we discuss box splines and quasi-interpolation to deal with multidimensional computations. In §2.5, we discuss some numerical aspects of our algorithm and give the error analysis. In §2.6, we summarize the constructions of quasi-interpolants which is needed in the implementation of our algorithm. In §2.7, we present some examples in one dimension to compare with the *algorithme à trous*. Some multidimensional examples are presented in the last section.

### §2.1. Computational Problems and *algorithme à trous*

To present the algorithm, we first introduce the following notations. For functions defined on $R^n$, we define: **Reflection:** $\iota f(t) = f(-t)$; **Shift:** $\tau_b f(t) = f(t-b)$; **Dilation:** $\sigma_a f(t) = f(t/a), a > 0$; **Convolution:** $K_f g = (f * g)(t) = (g * f)(t)$; and **Sampling:** $S_t f(n) = f(nh), h \in R, h > 0$. For sequences defined on $Z^n$, we define: **Translation:** $(T_\zeta a)(n) = a(n-\zeta)$, where $\zeta \in Z^n$; and we denote $(Ta)(n) = a(n-1)$ in the one dimensional case; **Dilation:**

$$D_p a(n) = \begin{cases} a(n/p), & n \equiv 0(mod\ p) \\ 0, & n \not\equiv 0(mod\ p) \end{cases}$$

where $p$ is a positive integer; **Convolution:** $K_a b = a * b = b * a$; **Local Complexity:** $\mathcal{L}(a) = \sum_{a(n)\neq 0} 1$; and $\delta_k$ the Kronecker delta function, $\delta_k(j) = 1$, if $j = k$ and $0$ otherwise.

Throughout this chapter, we set $\Psi(t) = \iota\psi(t)$, where $\psi(t)$ is the analyzing wavelet. Then the continuous wavelet transform is defined as:

$$\mathcal{T}f(b,a) = a^{-\frac{n}{2}} K_f(\sigma_a \iota\psi) = a^{-\frac{n}{2}} K_f \sigma_a \Psi.$$

For a given scale $a$, we get the following discrete version of the continuous wavelet transform:

$$\mathcal{T}^*f(ih,a) = a^{-\frac{n}{2}}h^n\sum_{j\in Z^n} f(jh)\Psi(\frac{(i-j)h}{a});$$

in other words,

$$\{\mathcal{T}^*f(ih,a)\} = a^{-\frac{n}{2}}h^n\{S_h f\} * \{S_h\sigma_a\Psi\}.$$

Since only dyadic scales are used in most applications, we are mainly interested in computing the transform at scales: $a, 2a, \cdots, 2^N a$. That is, we need the following sequences:

$$\text{scale } 2^i a\text{: } S_h f * S_h\sigma_{2^i a}\Psi, \quad 0 \leq i \leq N.$$

Since the key operation here is convolution, the **complexity** of the computation, as defined in [19], is determined by the local complexity of the convolution sequence:

$$|K_a| = \mathcal{L}(a).$$

Then $|K_a K_b| = |K_a| + |K_b| = \mathcal{L}(a) + \mathcal{L}(b)$.

Without loss of generality, we assume throughout this chapter that $a = 1$. Now let's look at a naive computation of the continuous wavelet transform. Suppose that the wavelet $\Psi(t)$ is compactly supported. Then

$$|K_{S_h\sigma_{2^N}\Psi}| = \mathcal{L}(S_h\sigma_{2^N}\Psi) \approx 2^{Nn}\mathcal{L}(S_h\Psi)$$

which tells us that the complexity is exponentially increasing. Therefore the naive computation will not be realistic if we want to compute the transform at several different scales.

The problem with the direct computation can be reasoned as follows. For general analyzing wavelet $\Psi(t)$, normally there is no easy relation between values $\{\Psi(\frac{j}{2}h)\}$ and $\{\Psi(jh)\}$, that is, $\{\Psi(\frac{j}{2}h)\}$ which is needed in the next scale cannot be easily generated from $\{\Psi(jh)\}$. This results in a lot of repeated computation. As the number of sampled wavelet values increases, the convolution will keep growing in an exponential manner. To improve upon this situation, we try to use the values at $\{\Psi(jh)\}$ to generate approximate values at $\{\Psi(jh/2)\}$ when some component of $j$ is odd, that is, to fill the "hole". With an easy relation between two consecutive levels, the long convolution can be factorized into a series of small convolutions. In the one dimensional case, this idea leads to the *algorithme à trous*.

**The algorithme à trous:** For a filter $F \in l_2(Z)$, if $O = (D_2 + TD_2K_F)$, then the continuous wavelet transforms are computed in the following way:

$$\text{scale } 2^i\text{: } S_hf * O^iS_h\Psi,\ 0 \le i \le N.$$

The following result by Holschneider, Kronland-Martinet, Morlet and Tchamitchian [19] tells the efficiency of the *algorithme à trous*.

**Theorem 1.1.** *Suppose $G, F \in l_2(Z)$ and $O$ is defined as: $O = D_2 + TD_2K_F$, then*

$$K_{O^NG} = K_{G_N}K_{F_1}\cdots K_{F_N},$$

*with $F_1 = \delta_0 + TD_2F$, $F_{i+1} = D_2F_i$ and $G_N = {D_2}^NG$. Moreover,*

$$|K_{G_N}K_{F_1}\cdots K_{F_N}| = \mathcal{L}(G) + N[1 + \mathcal{L}(F)].$$

**Implementation of the *algorithme à trous*:**

$$\begin{array}{llll}
\text{scale 1:} & & K_{S_h\Psi}f; & \\
\text{scale 2:} & F_1 = \delta_0 + TD_2F, & X_1 = K_{F_1}f, & \\
 & g_1 = D_2S_h\Psi, & K_{g_1}X_1; & \\
\text{scale } 2^j\text{:} & F_j = D_2F_{j-1}, & X_j = K_{F_{j-1}}X_{j-1}, & \\
 & g_j = D_2g_{j-1}, & K_{g_j}X_j; & \text{for } 2 \le j \le N.
\end{array}$$

**Examples 1.2.** For the filter $F(0) = 1$, $F(i) = 0$, if $i \neq 0$, we have

$$(O\Psi)(j) = \begin{cases} \Psi(\frac{j}{2}), & j \ even \\ \Psi(\frac{j-1}{2}), & j \ odd \end{cases}$$

This provides us with a piecewise constant approximation. Notice that $F_1 = \delta_0 + \delta_1$.

**Examples 1.3.** For the filter $F(0) = F(1) = \frac{1}{2}$, $F(i) = 0$, if $i \neq 0, 1$, we have

$$(O\Psi)(j) = \begin{cases} \Psi(\frac{j}{2}), & j \ even \\ \{\Psi(\frac{j-1}{2}) + \Psi(\frac{j+1}{2})\}/2, & j \ odd \end{cases}$$

This provides us with a piecewise linear approximation. Notice that $F_1 = \frac{1}{2}\delta_0 + \delta_1 + \frac{1}{2}\delta_2$.

Mathematically, the *algorithme à trous* uses an auxiliary function $g$ such that:

$$O^j S_h g = S(\sigma_2)^j g, \text{ for } j \geq 0$$

as the approximation of the analyzing wavelet $\Psi$. The function $g(t)$ also interpolates $\Psi(t)$:

$$g(jh) = \Psi(jh), \text{ for all } j \in Z$$

The two examples given are piecewise constant and linear approximations. A natural question is whether higher order splines can be applied in this approximation. Bounded by the construction of the filter $F$ and the above equations, the *algorithme à trous* cannot use higher order spline interpolations. The reason is that the *algorithme à trous* requires cardinal interpolation of $g$ to $\Psi$, but for the cardinal interpolation splines, the filter $F$ cannot be finite except when the spline order is less than two. To get smoother approximation to the analyzing wavelet, $F$ must be constructed very carefully. As Deslauriers and Dubuc's work [12][13] shows, the hole filling process will normally end up with fractal curves and the smoothness of the generated function remains yet to be studied.

## §2.2. Compute CWT By Subdivision

Instead of insisting on hole filling, we study the approximation of the analyzing wavelets by linear combination of the shifts of a single refinable function. The refinability provides us with an efficient subdivision scheme to replace hole filling. The approach also applies to computation in higher dimensions.

Suppose $\phi$ is a refinable function, that is, it satisfies the **refinement equation**

$$\phi(t) = \sum_{k \in Z^n} b(k)\phi(2t - kh),$$

with **refinement mask** $b$. Suppose further that all of the h-shifts of $\phi$ form a Riesz basis of $L_2(R^n)$, that is,

$$C_1 ||a||_2 \leq ||\sum_{j \in Z^n} a_j \phi(t - jh)||_2 \leq C_2 ||a||_2,$$

for any $a \in l_2(Z^n)$, with fixed constants $C_1$ and $C_2$, such that $0 < C_1 < C_2$. Let

$$c(j) = \phi(jh), \quad j \in Z^n.$$

This sequence connects the discrete values of $\{g(jh)\}$ with the coefficients $\{d(j)\}$ of the approximation by h-shifts of the function $\phi$. With the commutability of convolution and subdivision, we gain an efficient algorithm to compute CWT.

The idea of subdivision originated in the study of computer aided geometric design and corner-cutting algorithms. For a reference, one may refer to Cavaretta, Dahmen and Micchelli [4] and the references listed there. The following elementary results about subdivision are needed for the presentation of the new algorithm. We give very easy proofs to correspond with our notational setting and make our discussions complete.

**Lemma 2.1.** *If* $g(t) = \sum_{k \in Z^n} d(k)\phi(t - kh)$, *then* $S_h g = d * c$.

Proof: $S_h g(j) = g(jh) = \sum_{k\in Z^n} d(k)\phi(jh - kh) = d * c$. □

**Lemma 2.2.** *(Subdivision) If* $g(t) = \sum_{k\in Z^n} d(k)\phi(t - kh) = \sum_{k\in Z^n} dd(k)\phi(2t - kh)$, *then* $dd = D_2 d * b$.

Proof: According to the refinability of $\phi(t)$,

$$\begin{aligned} g(t) &= \textstyle\sum_{k\in Z^n} d(k)\phi(t - kh) \\ &= \textstyle\sum_{k\in Z^n} d(k)\sum_{j\in Z^n} b(j)\phi(2t - 2kh - jh) \\ &= \textstyle\sum_{k\in Z^n} d(k)\sum_{p\in Z^n} b(p - 2k)\phi(2t - ph) \\ &= \textstyle\sum_{p\in Z^n}\sum_{k\in Z^n} b(p - 2k)d(k)\phi(2t - ph) \\ &= \textstyle\sum_{p\in Z^n} dd(p)\phi(2t - ph). \end{aligned}$$

From the linear independence of $\{\phi(t - kh)\}$, we get

$$\begin{aligned} dd(p) &= \textstyle\sum_{k\in Z^n} b(p - 2k)d(k) \\ &= \textstyle\sum_{k\in Z^n} b(p - 2k)D_2 d(2k) \\ &= \textstyle\sum_{k\in Z^n} b(p - k)D_2 d(k), \end{aligned}$$

which gives us $dd = D_2 d * b$. □

**Lemma 2.3.** *If* $g(t) = \sum_{j\in Z^n} dd(j)\phi(2t - jh)$, *then* $S_h\sigma_2 g = dd * c$.

Proof: $S_h\sigma_2 g(j) = g(jh/2) = \sum_{k\in Z^n} dd(k)\phi(jh - kh)$, that is, $S_h\sigma_2 g = dd * c$. □

**Lemma 2.4.** *For any two sequence* $p$ *and* $q$, $D_2(p * q) = D_2 p * D_2 q$.

Proof: We just need to look at even integers. If $j = 2k$, then

$$\begin{aligned} D_2(p * q)(j) &= (p * q)(k) \\ &= \textstyle\sum_{l\in Z^n} p(l)q(k - l) \\ &= \textstyle\sum_{l\in Z^n} D_2 p(2l) D_2 q(2k - 2l) \\ &= \textstyle\sum_{l\in Z^n} D_2 p(2l) D_2 q(j - 2l), \end{aligned}$$

since $D_2p(i) = D_2q(i) = 0$, if $i \not\equiv 0\ mod(2)$, therefore,

$$D_2(p * q)(j) = \sum_{l \in Z^n} D_2p(l)D_2q(j-l)$$

which gives what we want. □

Finally, as a direct consequence of the definitions, we have

**Lemma 2.5.** *$\mathcal{L}(D_2^k a) = \mathcal{L}(a)$, for all $k \geq 0$.*

These preliminary results imply easily our main result.

**Theorem 2.6.** *If $g(t) = \sum_{j \in Z^n} d(j)\phi(t - jh)$, then*

$$S_h\sigma_2^k g = b * D_2 b * \cdots * D_2^{k-1} b * D_2^k d * c, \quad k \geq 0.$$

Proof: By induction, the conclusion when $k = 0$ is guaranteed by Lemma 2.1. Suppose the conclusion is true up to k, then according to Lemmas 2.2 and 2.3, $S_h\sigma_2^{k+1}g = D_2(b * D_2b * \cdots * D_2^{k-1}b * D_2^k d) * b * c$. Lemma 2.4 and the commutability of convolution give the results we want. □

Suppose $\Psi(t)$ is the given analyzing wavelet and let $g(t) = \sum_{j \in Z^n} d(j)\phi(t - jh)$ be a good approximation of $\Psi(t)$. Then the continuous wavelet transform can be approximated by

$$\mathcal{T}^0 f(ih, a) = a^{-\frac{n}{2}} h^n \sum_{j \in Z^n} f(jh) g(\frac{(i-j)h}{a})$$

or equivalently,

$$\{\mathcal{T}^0 f(ih, a)\} = a^{-\frac{n}{2}} h^n \{S_h f\} * \{S_h \sigma_a g\}$$

which is similar to the *algorithme à trous.* Naturally we require that $g(t)$ approximate $\Psi(t)$ well. The error introduced by this approximation will be studied in the following sections. We will see that as long as $g$ approximates $\Psi$ well in the $L_p$-norm, the numerical error will be small. Therefore the interpolatory condition

$$g(jh) = \Psi(jh), \quad j \in Z^n$$

is not necessary when applying this algorithm.

Following the previous discussions, we compute the continuous wavelet transform at dyadic scales by

$$\text{scale } 2^j\text{: } S_hf * S_h(\sigma_2)^j g,\ 0 \leq j \leq N.$$

According to Theorem 2.6, we summarize this approach as an algorithm.

**Algorithm 2.7.** Compute the continuous wavelet transform by

$$\text{scale } 2^j\text{: } S_hf * b * D_2b * D_2^2b * \cdots D_2^{j-1}b * D_2^jd * c,\ 0 \leq j \leq N.$$

**Implementation of Algorithme 2.7:** let $ff = S_hf * c$,

$$\begin{array}{llll}
\text{scale } 1\text{:} & g = d, & K_g ff; & \\
\text{scale } 2\text{:} & F_1 = b, & X_1 = K_{F_1} ff, & \\
 & g_1 = D_2 g, & K_{g_1} X_1; & \\
\text{scale } 2^j\text{:} & F_j = D_2 F_{j-1}, & X_j = K_{F_{j-1}} X_{j-1}, & \\
 & g_j = D_2 g_{j-1}, & K_{g_j} X_j; & \text{for } 2 \leq j \leq N.
\end{array}$$

Notice that the implementation pattern of our algorithm is almost identical to the *algorithme à trous*. The difference is that we substitute interpolation by quasi-interpolation and thus provide higher order approximation which leads to a more accurate computation of CWT. Refinability is required to apply the subdivision scheme. Viewing these two schemes in the one dimensional case, we see that the sequence $F_1$ here in this scheme is the refinement mask and thus can be understood more clearly.

Indeed, our algorithm coincides with the *algorithme à trous* if the approximation is by interpolation for spline spaces of order 1 and 2. Therefore the algorithm can be considered as an extension of the *algorithme à trous*.

**Theorem 2.8.** *If* $g(t) = \sum_{j \in Z^n} d(j)\phi(t - jh)$, *then* $K_{S_h \sigma_2^k g} = K_{b * D_2 b * \cdots * D_2^{k-1} b * D_2^k d * c}$ *and*

$$|K_b K_{D_2 b} \cdots K_{D_2^{k-1} b} K_{D_2^k d} K_c| = \mathcal{L}(c) + k\mathcal{L}(b) + \mathcal{L}(d)$$

Proof: According to Theorem 2.1 and Lemma 2.5, $|K_b K_{D_2 b} \cdots K_{D_2^{k-1} b} K_{D_2^k d} K_c| = |K_b| + |K_{D_2 b}| + \cdots + |K_{D_2^{k-1} b}| + |K_c| + |K_{D_2^k d}| = \mathcal{L}(c) + k\mathcal{L}(b) + \mathcal{L}(d)$. □

This theorem shows the complexity of Algorithm 2.7 and we shall see that it is a fairly fast algorithm. In the one dimensional case, it has the same performance as the *algorithme à trous*. Its speed depends on the length of the mask sequence. Some details about implementation will be discussed in the last two sections.

## §2.3. Cardinal B-splines and Quasi-interpolation: Univariate Case

Now let us discuss how our approach applies to higher order spline approximations where the *algorithme à trous* cannot be used. In the special case when $\phi(t)$ is the cardinal B-spline of order $k$, the approximation of the analyzing wavelet $\Psi(t)$ can be given by the well-studied quasi-interpolation. First we give a brief overview of B-splines and quasi-interpolation. Recall that cardinal B-spline of order $k$ when $k > 1$ is defined as:

$$M_k(t) = M_{k-1} * \chi_{(0,h]}(t)$$

starting with $M_1(t) = \chi_{(0,h]}(t)$. The following theorem summarizes some of the basic properties of B-splines(cf [3][26]).

**Theorem 3.1.** *If $M_k(t)$ is cardinal B-spline of order $k$, with $k \geq 1$, then*

*1.* $M_k(t) \geq 0$, $suppM_k(t) = [0, kh]$.

*2.* $M_k(t) \in C^{k-2}(R)$ *and is a polynomial of order* $k-1$ *on* $(ih, (i+1)h)$.

*3.* $\sum_{j\in Z} M_k(t - jh) = 1$.

*4. Recurrence relation:*

$$M_k(t) = \tfrac{t}{(k-1)h} M_{k-1}(t) + \tfrac{kh-t}{(k-1)h} M_{k-1}(t-h).$$

The sequence $c = \{M_k(jh)\}$, can be computed easily by the recurrence relation, from which we also see that $c(j) = \delta_1(j)$ if and only if $k = 1, 2$. The cardinal B-spline is refinable and the refinement mask is finite, more precisely

$$M_k(t) = \sum_{j\in Z} b(j) M_k(2t - jh)$$

where

$$b(j) = 2^{-k+1}\binom{k}{j}, \; j = 0, 1, \cdots, k$$

and $b(j) = 0$ otherwise.

**Example 3.2.** When $k = 1$, the refinement mask sequence is: $b = \delta_0 + \delta_1$ and the sequence $c$ is : $c = \delta_1$. If $f(t) = \sum_{j\in Z} d(j) M_1(t - jh)$, then $d(j) = f((j+1)h)$.

**Example 3.3.** When $k = 2$, the refinement mask sequence is: $b(j) = \frac{1}{2}\delta_0 + \delta_1 + \frac{1}{2}\delta_2$ and the sequence $c$ is : $c = \delta_1$. If $f(t) = \sum_{j\in Z} d(j) M_2(t - jh)$, then $d(j) = f((j+1)h)$.

From Example 1.1 and Example 1.2, we see that our algorithm coincides with the *algorithme à trous* in these two cases. And these are the only cases in which they coincide as shown by the fact that $c = \delta_1$ is a delta function if and only if $k = 1, 2$. For characterization of compactly supported refinable splines, one may refer to Lawton, Lee and Shen [21]. These two algorithms take different approaches to generate the values at the finer mesh: filling holes or subdivision. These two techniques have been studied in other areas with a common purpose: render the given data and provide an approximation to the function which provides the discrete data. The above observation reveals the connection between these two approaches in the context of computation of CWT.

The next thing we need to check is the approximation order of the quasi-interpolation of B-splines. This approach has been well studied in approximation theory. Here we list the following result of Jia and Lei [20] to support our discussion of continuous wavelet transform computation.

**Theorem 3.4.** *If $\Psi \in C^k(R)$ and decays exponentially, then for $1 \leq p \leq \infty$, there exists $g(t) = \sum_{j \in Z} d(j) M_k(t - jh)$ which decays exponentially and which satisfies*

$$||\Psi - g||_p \leq C h^k$$

*for some positive constant $C$.*

## §2.4. Box Splines and Quasi-Interpolation: Multivariate Case

As a natural extension of cardinal B-splines to the multivariate case, box splines provide bases for spline approximation of functions in higher dimensions. For a given $h > 0$, we define (cf de Boor, Höllig and Riemenschneider [3]) a box spline $M_\Xi(t)$ corresponding to an invertible $n \times n$ integer matrix $\Xi$ by

$$M_\Xi = \frac{1}{|det \Xi|} \chi_{\Xi \Box_h}$$

where $\Box_h = (0, h]^n$. If $\Xi \bigcup \zeta$ is a matrix formed from $\Xi$ by the addition of the column $\zeta \in Z^n$, then

$$M_{\Xi \bigcup \zeta} = \int_0^h M_\Xi(\cdot - t\zeta) dt.$$

Thus we can define the box spline $M_\Xi$ for any $n \times s$ integer matrix $\Xi$ with rank $n$. This definition differs from the standard one [3] by the introduction of the scaling factor $h$ and box splines can be defined corresponding to any $n \times s$ matrix $\Xi$ [3].

Set $k = s - n + 1$ and define

$$m = \min\{\#Z : rank(\Xi \setminus Z) < n\}$$

where $\Xi \setminus Z$ denotes the matrix obtained by removing the columns from $Z$ and $\#Z$ is the number of columns in matrix $Z$. The box spline $M_\Xi$, as defined above, has some nice properties. We summarize them into the following Theorem(cf [3]).

**Theorem 4.1.** *If $\Xi$ is a $n \times s$ integer matrix which has rank $n$, then*

1. $M_\Xi(t) \geq 0, suppM_\Xi = \Xi\square_h$.
2. $M_\Xi \in C^{m-2}(R^n)$ *and is piecewise polynomial of degree* $m - 1$.
3. $\sum_{j\in Z^n} M_\Xi(t - jh) = 1$.
4. *Recurrence relation: if* $M_{\Xi\setminus\xi}, \xi \in \Xi$ *are continuous at* $x = \Xi t$, *then*

$$(s-n)hM_\Xi(x) = \sum_{\xi\in\Xi}\{t_\xi M_{\Xi\setminus\xi}(x) + (h - t_\xi)M_{\Xi\setminus\xi}(x - \xi h)\}.$$

The sequence $c(j) = M_\Xi(jh), j \in Z^n$ can be computed by the recurrence relation. The box spline $M_\Xi$ is refinable and the mask sequence is finite. More precisely(cf [3]),

**Theorem 4.2.** *If $\zeta_1, \zeta_2, \cdots, \zeta_s$ are the $s$ columns of a $n \times s$ integer matrix $\Xi$ of rank $n$, then box spline $M_\Xi$ satisfies*

$$M_\Xi(t) = \sum_{j\in Z^n} b(j) M_\Xi(2t - jh)$$

*where the mask sequence can be computed as*

$$b = 2^{(s-n)}\delta_0 * [\delta_0 + \delta_{\zeta_1}] * [\delta_0 + \delta_{\zeta_2}] * \cdots * [\delta_0 + \delta_{\zeta_s}].$$

As for the quasi-interpolation of box splines, following Jia and Lei [20], we have he following approximation order.

**Theorem 4.3.** *If $\Psi \in C^k(R^n)$ and decays exponentially, then for $1 \le p \le \infty$ and an $n \times s$ integer matrix $\Xi$ of rank $n$, there exist $g(t) = \sum_{j \in Z^n} d(j) M_\Xi(t - jh)$ which decays exponentially and which satisfies*

$$\|\Psi - g\|_p \le Ch^m$$

*where $C > 0$ is a positive constant.*

The connection between cardinal B-splines and box splines is that when $\Xi = [1, 1, \cdots, 1]$ is a $1 \times k$ matrix, then $M_\Xi(t) = M_k(t)$. And $m$ in this case is $k$.

## §2.5. Error Analysis

In our algorithm, we use an auxiliary function which is a linear combination of shifts of a refinable function to approximate the analyzing wavelet $\Psi(t)$. Numerical error is thus introduced by this approximation. Here we present some basic error analysis for this algorithm.

According to the previous definitions, we have, for all $j \ge 0$,

$$\begin{aligned} & \{\mathcal{T}^0 f(ih, 2^j)\} - \{\mathcal{T}^* f(ih, 2^j)\} \\ = \;& 2^{-\frac{jn}{2}} h^n (S_h f) * (S_h \sigma_2^j \Psi) - h^n (S_h f) * (S_h \sigma_2^j g) \\ = \;& 2^{-\frac{jn}{2}} h^n (S_h f) * S_h \sigma_2^j (\Psi - g) \\ = \;& 2^{-\frac{jn}{2}} h^n (S_h f) * S_{h/2^j} (\Psi - g). \end{aligned}$$

As we compute the continuous convolution by means of a discrete convolution, we need first some results which measures the error of this first step.

**Lemma 5.1.** *Suppose $g$ is continuous and has exponential decay, then*

$$||S_h g||_p \leq C h^{-n/p} ||g||_p,$$

*for $1 \leq p \leq \infty$ and here $C$ is a positive constant.*

Proof: If $||g||_p = 0$, then $g(t) \equiv 0$ which will guarantee our conclusion. For $||g||_p > 0$, we gain our conclusion by considering the Riemann Sum of integration, since our conditions guarantee that $g(t)$ is also Riemann summable. □

**Theorem 5.2.** *Suppose $\Psi$ and $g$ are continuous and have exponential decay, where $g(t) = \sum_{k \in Z^n} d(k)\phi(t - kh)$, and*

$$||\Psi - g||_p \leq C h^m,$$

*for some $1 \leq p \leq \infty$, $m \geq 0$ and with constant $C > 0$. Then*

$$||S_{h/2^j}(\Psi - g)||_p \leq C 2^{jn/p} h^{m-n/p}$$

*Here $C$ is a positive constant.*

Proof: From Lemma 5.1 and the given conditions, we get:

$$\begin{aligned} & ||S_{h/2^j}(\Psi - g)||_p \\ \leq \quad & C 2^{jn/p} |h|^{-n/p} ||\Psi - g||_p \\ \leq \quad & C 2^{jn/p} |h|^{-n/p} |h|^m \\ \leq \quad & C 2^{jn/p} |h|^{m-n/p} \end{aligned}$$

which gives the result we want. □

Note that the estimate in Theorem 5.2 cannot be improved generally, since normally the estimates in Lemma 4.1 are sharp. We can see from these basic estimates in order that to reduce the order of the truncation error, higher order approximation is needed; particularly when we need to compute CWT at quite a few scales. Turning to the truncation error, from the structure of our algorithm and the previous results, we get the following conclusion.

**Theorem 5.3.** *Suppose $\Psi$ and $g$ are continuous and have exponential decay, where $g(t) = \sum_{k\in Z^n} d(k)\phi(t - kh)$, and*

$$\|\Psi - g\|_p \le Ch^m,$$

*for $1 \le p \le \infty$, $m \ge 0$ and with constant $C > 0$. Then*

$$\|T^*f(ih,2^j) - T^0f(ih,2^j)\|_1 \le C2^{\frac{jn}{2}}h^m\|S_hf\|_1,$$

$$\|T^*f(ih,2^j) - T^0f(ih,2^j)\|_\infty \le C2^{-\frac{jn}{2}}h^{m+n}\|S_hf\|_1,$$

$$\|T^*f(ih,2^j) - T^0f(ih,2^j)\|_\infty \le C2^{\frac{jn}{2}}h^m\|S_hf\|_\infty.$$

*Here $C$ is a positive constant and $1 \le p \le \infty$.*

Proof: By the definition of $T^*$ and $T^0$, with the help of Theorem 5.2, we get

$$\begin{aligned}
& \|T^0f(ih,2^j) - T^*f(ih,2^j)\|_1 \\
\le \quad & \|2^{-\frac{jn}{2}}h^n(S_hf) * S_{h/2^j}(\Psi - g)\|_1 \\
\le \quad & 2^{-\frac{jn}{2}}h^n\|S_{h/2^j}(\Psi - g)\|_1\|S_hf\|_1 \\
\le \quad & C2^{-\frac{jn}{2}}h^n2^{jn}h^{m-n}\|S_hf\|_1 \\
\le \quad & C2^{\frac{jn}{2}}h^{m-n}\|S_hf\|_1.
\end{aligned}$$

Other two inequalities can be proved in the same way. □

**Remark 5.4.** Let $\phi$ be the B-spline of order $k$ or the box spline $M_\Xi$ with an $n \times s$ integer matrix of rank $n$, let $m = \min\{\#Z : rank(\Xi \setminus Z) < n\}$ where $\#Z$ is the number of columns in matrix $Z$. Suppose the analyzing wavelet $\Psi(t) \in C^m(R^n)$ and decays exponentially, the quasi-interpolant $g(t) = \sum_{j\in Z^n} d(j) M_\Xi(t - jh)$ exponentially decays, and

$$||S_{h/2^j}(\Psi - g)||_p \le C2^{jn/p}h^{m-n/p}$$

where $C$ is a positive constant and $1 \le p \le \infty$. Therefore we have

$$||\mathcal{T}^* f(ih, 2^j) - \mathcal{T}^0 f(ih, 2^j)||_1 \le C2^{\frac{jn}{2}} h^m ||S_h f||_1,$$

$$||\mathcal{T}^* f(ih, 2^j) - \mathcal{T}^0 f(ih, 2^j)||_\infty \le C2^{-\frac{jn}{2}} h^{m+n} ||S_h f||_1,$$

$$||\mathcal{T}^* f(ih, 2^j) - \mathcal{T}^0 f(ih, 2^j)||_\infty \le C2^{\frac{jn}{2}} h^m ||S_h f||_\infty,$$

where $C$ is a positive constant.

It is clear that higher order accuracy is needed, then either higher order approximation or increased sampling frequency (that is, smaller $h$) is necessary. The higher order approximation improves accuracy within a given sampling frequency without much cost.

## §2.6. Implementation With Quasi-Interpolation

Quasi-interpolation originated in finite element analysis and approximation theory. One may refer to Strang and Fix [28], Dahmen and Micchelli [9], de Boor [2] and Chui and Diamond [6] for details. From the discussion of the last section, we know quasi-interpolation can provide higher order approximation, and thus leads to more accurate computations. The realization of quasi-interpolation is typically done as follows. For a given $n \times s$ integer matrix $\Xi$ with rank $n$, denote $m = \min\{\#Z : rank(\Xi \setminus Z) < n\}$. If we have a bounded linear functional $\lambda$ such that the operator

$$Qf(t) = \sum_{j\in Z^n} M_\Xi(t - jh)\lambda f(h \cdot + jh)$$

reproduces polynomials of order $m-1$, then for $f(t) \in C^m(R)$ with exponential decay, we have

$$||Qf - f||_p \le Ch^m$$

where $C$ is a positive constant and $0 \le p \le \infty$.

Therefore it remains to find the right functional $\lambda$ to get the quasi-interpolants of certain approximation order. Actually there are several explicit construction schemes to do this. Following [3], we present two basic approaches of this construction.

**Approach I.** The functional $\lambda$ is given by

$$\lambda f = \sum\nolimits_{0\le|\alpha|<m} h^{|\alpha|} g_\alpha(0)(\mathcal{D}^\alpha f)(0)$$

where $\mathcal{D}$ is a differential operator and $g_\alpha(t)$ is a polynomial of degree $|\alpha|$. These polynomials can be generated by induction with

$$g_\alpha(t) = \frac{t^\alpha}{\alpha!} - \sum\nolimits_{0\le\beta<\alpha} \frac{\mu t^{\alpha-\beta}}{(\alpha-\beta)!} g_\beta(t),$$

which starts with $g_0(t) = 1$ and where

$$\mu f = \sum\nolimits_{l\in Z^n} M_\Xi(lh) f(-l).$$

Notice that $\{M_\Xi(lh)\}$ is the sequence $c$ we used in our algorithm. Thus the quasi-interpolant can be constructed explicitly according to the box spline used.

**Approach II.** Notice that derivatives up to order $m-1$ are used in the previous approach. We can also construct the functional $\lambda$ without using the derivatives. Such a $\lambda$ can be constructed as

$$\lambda f = \sum\nolimits_{j\in Z^n} \omega(j) f(-jh)$$

where $\omega$ is generated by

$$\omega = \delta_0 + v + v * v + \cdots + \underbrace{v * \cdots * v}_{m-1}$$

and $v$ is defined by

$$v = \delta_0 - \sum_{j \in Z^n} M_k(jh)\delta_j.$$

Clearly we have

$$d(i) = \lambda f(h \cdot +ih) = \sum_{j \in Z} \omega(j) f(ih - jh),$$

that is,

$$d = \omega * S_h f.$$

Following this approach, once $\omega$ is computed, we can implement our algorithm as:

**Algorithme 6.2:** let $ff = S_h f * c$,

```
scale 1:     $g = \omega * S_h \Psi$,          $K_g ff$;
scale 2:     $F_1 = b$,                $X_1 = K_{F_1} ff$,
             $g_1 = D_2 g$,             $K_{g_1} X_1$;
scale $2^j$:    $F_j = D_2 F_{j-1}$,      $X_j = K_{F_{j-1}} X_{j-1}$,
             $g_j = D_2 g_{j-1}$,       $K_{g_j} X_j$;          for $2 \le j \le n$.
```

## §2.7. Numerical Examples: Univariate Case

First we look at the one dimensional case where we consider the commonly used Mexican Hat wavelet

$$\psi(t) = (1 - t^2)e^{-\frac{t^2}{2}}.$$

Notice that $\Psi = \psi$ in this case. Following the two schemes given in §2.6, we can construct the quasi-interpolants explicitly.

**Example 7.1.** For the linear B-spline $M_2(t)$, by Approach I, we have $g_0(0) = 1$ and $g_1(0) = 1$. Thus the functional $\lambda$ given by Approach I is defined as

$$\lambda f = f(0) + f'(0)h.$$

For the Mexican Hat, the quasi-interpolant of approximation order 2 has coefficient sequence $d$ given by

$$d(j) = \{1-(jh)^2\}e^{-\frac{(jh)^2}{2}} + h\{(jh)^3 - 3(jh)\}e^{-\frac{(jh)^2}{2}}, \quad j \in Z.$$

Notice that for the **algorithme à trous**, we need interpolation. Here we can also give up the interpolation requirement and use quasi-interpolation. They both have approximation order of 2 and the only difference in computation is that different sequence $d$ is used.

For the quadratic B-spline $M_3(t)$, we have $g_0(0) = 1$, $g_1(0) = 3/2$ and $g_2(0) = 1$. Therefore the $\lambda$ in Approach I is defined as

$$\lambda f = f(0) + \frac{3}{2}f'(0)h + f''(0)h^2.$$

For the Mexican Hat, the quasi-interpolant of approximation order 3 has coefficient sequence $d$ given by

$$\begin{aligned} d(j) = \quad & \{1-(jh)^2\}e^{-\frac{(jh)^2}{2}} \\ & +\tfrac{3h}{2}\{(jh)^3 - 3(jh)\}e^{-\frac{(jh)^2}{2}} \\ & +h^2\{-(jh)^4 + 6(jh)^2 - 3\}e^{-\frac{(jh)^2}{2}}. \end{aligned}$$

For the most commonly used B-spline, the cubic spline $M_4(t)$, we have $g_0(0) = 1$, $g_1(0) = 2$, $g_2(0) = 11/6$ and $g_3(0) = 1$. Therefore the $\lambda$ given by Approach I is defined as

$$\lambda f = f(0) + 2f'(0)h + \frac{11}{6}f''(0)h^2 + f^{(3)}(0)h^3.$$

For the Mexican Hat, the quasi-interpolant of order 4 has the coefficients $d(j)$ given by

$$
\begin{aligned}
d(j) = \quad & \{1-(jh)^2\}e^{-\frac{(jh)^2}{2}} \\
& +2h\{(jh)^3-3(jh)\}e^{-\frac{(jh)^2}{2}} \\
& +\tfrac{11h^2}{6}\{-(jh)^4+6(jh)^2-3\}e^{-\frac{(jh)^2}{2}} \\
& +h^3\{(jh)^5-10(jh)^3+15(jh)\}e^{-\frac{(jh)^2}{2}}.
\end{aligned}
$$

**Example 7.2.** When applying Approach II, we get the following computation of the quasi-interpolants of the Mexican Hat. For the linear B-spline $M_2(t)$, we have

$$v = \delta_0 - \delta_1,$$

and $\omega$ is computed as

$$\omega = 2\delta_0 - \delta_1.$$

For the quadratic $M_3(t)$, we have

$$v = \delta_0 - \frac{1}{2}\delta_1 - \frac{1}{2}\delta_2,$$

and $\omega$ is computed as

$$\omega = 3\delta_0 - \frac{3}{2}\delta_1 - \frac{5}{4}\delta_2 + \frac{1}{2}\delta_3 + \frac{1}{4}\delta_4.$$

For the cubic spline $M_4(t)$, we have

$$v = \delta_0 - \frac{1}{6}\delta_1 - \frac{2}{3}\delta_2 - \frac{1}{6}\delta_3,$$

and $\omega$ turns out to be

$$\omega = 4\delta_0 - \delta_1 - \frac{35}{9}\delta_2 - \frac{25}{216}\delta_3 + \frac{35}{18}\delta_4 + \frac{47}{72}\delta_5 - \frac{8}{27}\delta_6 - \frac{17}{72}\delta_7 - \frac{1}{18}\delta_8 - \frac{1}{216}\delta_9.$$

**Example 7.3.** Following our error discussions in §2.5, we have

$$
\begin{aligned}
& \{T^0 f(ih,2^j)\} - \{T^* f(ih,2^j)\} \\
= \; & 2^{-\frac{jn}{2}} h^n (S_h f) * (S_h \sigma_2^j \Psi) - h^n (S_h f) * (S_h \sigma_2^j g) \\
= \; & 2^{-\frac{jn}{2}} h^n (S_h f) * S_h \sigma_2^j (\Psi - g) \\
= \; & 2^{-\frac{jn}{2}} h^n (S_h f) * S_{h/2^j} (\Psi - g).
\end{aligned}
$$

For a given signal $f$, the computational errors are mainly determined by

$$2^{-\frac{j}{2}} S_{h/2^j}(\Psi - g).$$

Let $E(j) = 2^{-\frac{j}{2}} ||S_{h/2^j}(\Psi - g)||_{\infty}$, Table 7.4 and Table 7.5 show the approximation error of the Mexican Hat by using Approach I with $h = 0.1$ and $h = 0.05$ respectively. Table 7.6 show and Table 7.7 show the approximation error of the Mexican Hat by using Approach II with $h = 0.1$ and $h = 0.05$ respectively. We can see from these experiments that higher order approximation is beneficial. We also observe that Approach I is better than Approach II in these computations. In the linear case, we have three different implementations: the linear *algorithme à trous*, quasi-interpolation by Approach I and quasi-interpolation by Approach II. In Table 7.8 and Table 7.9, we compare these three approaches with $h = 0.1$ and $h = 0.05$ respectively. Quasi-interpolations also give better results in this case.

**Table 7.4. Approach I:** $h = 0.1$

| $h = 0.1$ | Linear | quadratic | cubic |
|---|---|---|---|
| E(0) | 0.0149376 | 0.0021506 | 0.0005142 |
| E(1) | 0.0105625 | 0.0015236 | 0.0003657 |
| E(2) | 0.0074688 | 0.0010975 | 0.0002585 |
| E(3) | 0.0052812 | 0.0007760 | 0.0001830 |
| E(4) | 0.0037344 | 0.0005490 | 0.0001294 |
| E(5) | 0.0026406 | 0.0003882 | 0.0000915 |
| E(6) | 0.0018672 | 0.0002745 | 0.0000647 |
| E(7) | 0.0013203 | 0.0001941 | 0.0000457 |
| E(8) | 0.0009336 | 0.0001372 | 0.0000323 |
| E(9) | 0.0006601 | 0.0000970 | 0.0000228 |

**Table 7.5. Approach I:** $h = 0.05$

| $h = 0.05$ | Linear | quadratic | cubic |
|---|---|---|---|
| E(0) | 0.0037460 | 0.0002705 | 0.0000324 |
| E(1) | 0.0026488 | 0.0001913 | 0.0000231 |
| E(2) | 0.0018730 | 0.0001379 | 0.0000163 |
| E(3) | 0.0013244 | 0.0000975 | 0.0000115 |
| E(4) | 0.0009365 | 0.0000690 | 0.0000081 |
| E(5) | 0.0006622 | 0.0000488 | 0.0000057 |
| E(6) | 0.0004682 | 0.0000345 | 0.0000040 |
| E(7) | 0.0003311 | 0.0000244 | 0.0000028 |
| E(8) | 0.0002341 | 0.0000172 | 0.0000020 |
| E(9) | 0.0001655 | 0.0000122 | 0.0000014 |

**Table 7.6. Approach II:** $h = 0.1$

| $h = 0.1$ | linear | quadratic | cubic |
|---|---|---|---|
| E(0) | 0.0298752 | 0.0189414 | 0.0223314 |
| E(1) | 0.0211250 | 0.0134195 | 0.0158314 |
| E(2) | 0.0149376 | 0.0095165 | 0.0112041 |
| E(3) | 0.0105625 | 0.0067292 | 0.0079234 |
| E(4) | 0.0074688 | 0.0047582 | 0.0056031 |
| E(5) | 0.0052812 | 0.0033646 | 0.0039620 |
| E(6) | 0.0037344 | 0.0023791 | 0.0028015 |
| E(7) | 0.0026406 | 0.0016823 | 0.0019810 |
| E(8) | 0.0018672 | 0.0011895 | 0.0014007 |
| E(9) | 0.0013203 | 0.0008411 | 0.0009905 |

**Table 7.7. Approach II:** $h = 0.05$

| $h = 0.05$ | linear | quadratic | cubic |
|---|---|---|---|
| E(0) | 0.0074921 | 0.0024226 | 0.0014729 |
| E(1) | 0.0052977 | 0.0017130 | 0.0010424 |
| E(2) | 0.0037460 | 0.0012141 | 0.0007372 |
| E(3) | 0.0026488 | 0.0008585 | 0.0005213 |
| E(4) | 0.0018730 | 0.0006071 | 0.0003686 |
| E(5) | 0.0013244 | 0.0004293 | 0.0002606 |
| E(6) | 0.0009365 | 0.0003035 | 0.0001843 |
| E(7) | 0.0006622 | 0.0002146 | 0.0001303 |
| E(8) | 0.0004682 | 0.0001517 | 0.0000921 |
| E(9) | 0.0003311 | 0.0001073 | 0.0000651 |

**Table 7.8. Linear Case Comparison:** $h = 0.1$

| $h = 0.1$ | algorithme à trous | Approach I | Approach II |
|---|---|---|---|
| E(0) | 0 | 0.0149376 | 0.0298752 |
| E(1) | 0.0487458 | 0.0105625 | 0.0211250 |
| E(2) | 0.0517037 | 0.0074688 | 0.0149376 |
| E(3) | 0.0426406 | 0.0052812 | 0.0105625 |
| E(4) | 0.0322978 | 0.0037344 | 0.0074688 |
| E(5) | 0.0235961 | 0.0026406 | 0.0052812 |
| E(6) | 0.0169528 | 0.0018672 | 0.0037344 |
| E(7) | 0.0120822 | 0.0013203 | 0.0026406 |
| E(8) | 0.0085768 | 0.0009336 | 0.0018672 |
| E(9) | 0.0060766 | 0.0006601 | 0.0013203 |

**Table 7.9. Linear Case Comparison:** $h = 0.05$

| $h = 0.05$ | algorithme à trous | Approach I | Approach II |
|---|---|---|---|
| E(0) | 0 | 0.0037460 | 0.0074921 |
| E(1) | 0.0243937 | 0.0026488 | 0.0052977 |
| E(2) | 0.0258651 | 0.0018730 | 0.0037460 |
| E(3) | 0.0213325 | 0.0013244 | 0.0026488 |
| E(4) | 0.0161598 | 0.0009365 | 0.0018730 |
| E(5) | 0.0118073 | 0.0006622 | 0.0013244 |
| E(6) | 0.0084837 | 0.0004682 | 0.0009365 |
| E(7) | 0.0060465 | 0.0003311 | 0.0006622 |
| E(8) | 0.0042923 | 0.0002341 | 0.0004682 |
| E(9) | 0.0030411 | 0.0001655 | 0.0003311 |

## §2.8. Numerical Examples: Multivariate Case

Multivariate quasi-interpolants can be constructed in a similar way as in the one dimensional case. Following Approach I and Approach II, we just need to find the proper functional $\lambda$. We consider the two dimensional case in this section. As a comparison the one dimensional case, we use the two dimensional Mexican Hat

$$\psi(x,y) = (1 - x^2 - y^2)e^{-\frac{x^2+y^2}{2}}$$

as our analyzing wavelet. We consider the two most commonly used two dimensional box splines [3] which correspond to the linear and the quadratic B-splines respectively in the one dimensional case.

**Example 8.1.** Consider the box spline $M_\Xi$ where

$$\Xi = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

with $m = \{\#Z : rank(\Xi \setminus Z) < 2\} = 2$. The sequence $c$ is now

$$c(j) = M_\Xi(jh) = \delta_{(1,1)}(j), \quad j \in Z^2.$$

When applying Approach I, the functional $\mu$ is given by

$$\mu f = f(-h, -h)$$

where the polynomials $g_\alpha$ are given by

$$g_{(0,0)}(t) = 1,$$

$$g_{(1,0)}(t) = t^{(1,0)} + 1,$$

$$g_{(0,1)}(t) = t^{(0,1)} + 1.$$

Thus the functional $\lambda$ is given by

$$\lambda f = f(0) + h\mathcal{D}^{(1,0)}f(0) + h\mathcal{D}^{(0,1)}f(0).$$

As for Approach II, we have

$$v(j) = \delta_{(0,0)} - \delta_{(1,1)}$$

and $\omega$ is simply

$$\omega = 2\delta_{(0,0)} - \delta_{(1,1)}.$$

**Remark:** Under an affine map $y = Pt - (\frac{1}{2}, \frac{\sqrt{3}}{2})$, where $P$ is

$$P = \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{bmatrix},$$

the function $M_\Xi(P^{-1}y + (1,1))$ is a piecewise linear function defined on a hexagonal mesh and has function value 1 at the origin.

**Example 8.2.** Consider the Zwart element, that is the box spline $M_\Xi$ where

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

with $m = \{\#Z : rank(\Xi \setminus Z) < 2\} = 3$. The sequence $c$ is now

$$c(j) = M_\Xi(jh) = \begin{cases} \frac{1}{4}, & j = (0,1),(1,1),(0,2),(1,2); \\ 0, & otherwise. \end{cases}$$

When applying Approach I, the functional $\mu$ is given by

$$\mu f = (f(0,-h) + f(-h,-h) + f(0,-2h) + f(-h,-2h))/4$$

and the polynomials $g_\alpha$ are

$$g_{(0,0)}(t) = 1,$$

$$g_{(1,0)}(t) = t^{(1,0)} + \tfrac{1}{2},$$

$$g_{(0,1)}(t) = t^{(0,1)} + \tfrac{3}{2},$$

$$g_{(1,1)}(t) = g_{(1,0)}(t)g_{(0,1)}(t)\ ,$$

$$g_{(2,0)}(t) = \tfrac{t^{(2,0)}}{2} + \tfrac{1}{2}t^{(1,0)},$$

$$g_{(0,2)}(t) = \tfrac{t^{(0,2)}}{2} + \tfrac{3}{2}t^{(0,1)} + 1.$$

Thus the functional $\lambda$ is given by

$$\lambda f = f(0) + \frac{h}{2}\mathcal{D}^{(1,0)}f(0) + \frac{3h}{2}\mathcal{D}^{(0,1)}f(0) + \frac{3h^2}{4}\mathcal{D}^{(1,1)}f(0) + h^2\mathcal{D}^{(0,2)}f(0).$$

As for Approach II, we have

$$v(j) = \delta_{(0,0)} - \frac{1}{4}(\delta_{(0,1)} + \delta_{(1,1)} + \delta_{(0,2)} + \delta_{(1,2)})$$

and $\omega$ turns out to be

$$\begin{aligned}\omega = \ & 3\delta_{(0,0)} - \tfrac{3}{4}\delta_{(1,0)} - \tfrac{3}{4}\delta_{(1,1)} - \tfrac{11}{16}\delta_{(0,2)} - \tfrac{5}{8}\delta_{(1,2)} + \tfrac{1}{16}\delta_{(2,2)} + \tfrac{1}{8}\delta_{(0,3)} \\ & + \tfrac{1}{4}\delta_{(1,3)} + \tfrac{1}{8}\delta_{(2,3)} + \tfrac{1}{16}\delta_{(0,4)} + \tfrac{1}{8}\delta_{(1,4)} + \tfrac{1}{16}\delta_{(2,4)}.\end{aligned}$$

**Example 8.3.** We can compute the quasi-interpolants of the Mexican Hat by using each of the box splines presented in the previous two examples with the guidance of the two approaches. Similar to the one dimensional case, we notice that

$$E(j) = 2^{-\frac{jn}{2}}||S_{h/2^j}(\Psi - g)||_\infty \approx 2^{-\frac{jn}{2}}||S_h(\Psi - g)||_\infty.$$

Thus we test $||S_h(\Psi - g)||_\infty$ by using different step size $h$. Call the box spline in Example 8.1 as $M_1$ and the box spline in Example 8.2 (that is Zwart-Element) as $M_2$. The first two columns of Table 8.4 show the error distribution of $E(0) = ||S_h(\Psi - g)||_\infty$ by using $M_1$ and the two approaches respectively. The seven rows of Table 8.4 correspond to $h$ = 0.4, 0.25, 0.2, 0.1, 0.05, 0.025, 0.0125 respectively. Thus we can compare the accuracy of these two approaches. In the last two columns of Table 8.4, we repeat the computations in the first two columns by using the Zwart-element, that is $M_2$. These computations demonstrate exactly the approximation order as predicted in theory.

In Table 8.5 and Table 8.6, we list the the subdivision error

$$E(j) = 2^{-\frac{jn}{2}}||S_{h/2^j}(\Psi - g)||_\infty$$

for $M_1$ and $M_2$ with $h$ = 0.4, 0.2 and by Approach I and Approach II respectively. We just list the results with $j$ = 0, 1, 2, 3, 4. These computations verify that

$$E(j) = 2^{-\frac{jn}{2}}||S_{h/2^j}(\Psi - g)||_\infty \approx 2^{-\frac{jn}{2}}||S_h(\Psi - g)||_\infty$$

and once again we observe that Approach I is more accurate than Approach II.

**Table 8.4. Error Distribution of $||S_h(\Psi - g)||_\infty$**

| $M_1$: Approach I | $M_1$: Approach II | $M_2$: Approach I | $M_2$: Approach II |
|---|---|---|---|
| 0.4205422 | 0.8410844 | 0.1193488 | 0.9189762 |
| 0.1780135 | 0.3560271 | 0.0323426 | 0.2986730 |
| 0.1160737 | 0.2321474 | 0.0167859 | 0.1592006 |
| 0.0297511 | 0.0595023 | 0.0021809 | 0.0222092 |
| 0.0074843 | 0.0149687 | 0.0002751 | 0.0028439 |
| 0.0018740 | 0.0037480 | 0.0000344 | 0.0003570 |
| 0.0004686 | 0.0009373 | 0.0000043 | 0.0000446 |

**Table 8.5. Error Distribution of $E(j)$ with $M_1$**

| Approach I | | Approach II | |
|---|---|---|---|
| $h = 0.4$ | $h = 0.2$ | $h = 0.4$ | $h = 0.2$ |
| 0.4205422 | 0.1160737 | 0.8410844 | 0.2321474 |
| 0.2102711 | 0.0580368 | 0.4205422 | 0.1160737 |
| 0.1051355 | 0.0290184 | 0.2102711 | 0.0580368 |
| 0.0525677 | 0.0145092 | 0.1051355 | 0.0290184 |
| 0.0205931 | 0.0067729 | 0.0347382 | 0.0120470 |

**Table 8.6. Error Distribution of $E(j)$ with $M_2$**

| Approach I | | Approach II | |
|---|---|---|---|
| $h = 0.4$ | $h = 0.2$ | $h = 0.4$ | $h = 0.2$ |
| 0.1193488 | 0.0167859 | 0.9189762 | 0.1592006 |
| 0.0608828 | 0.0086013 | 0.4615941 | 0.0818611 |
| 0.0305858 | 0.0043172 | 0.2323837 | 0.0409451 |
| 0.0152361 | 0.0021277 | 0.1167059 | 0.0204811 |
| 0.0076293 | 0.0009840 | 0.0570554 | 0.0102405 |

# REFERENCES

1. C. de Boor, A practical guide to splines, Springer-Verlag, New York, 1978.

2. C. de Boor, The polynomials in the linear span of integer translates of a compactly supported function, Constr. Approx., 3(1987), 199-203.

3. C. de Boor, K. Höllig and S. Riemenschneider, Box splines, Springer-Verlag, New York, 1993.

4. A. S. Cavaretta, W. Dahmen and C. Micchelli, Stationary subdivision, Memoirs Amer. Math. Soc. #453, 93(1991).

5. C. Chui, An introduction to wavelets, Academic Press, New York, 1992.

6. C. Chui and H. Diamond, A natural formulation of quasi-interpolation by multivariate splines, Proc. Amer. Math. Soc., 99(1987), 643-646.

7. R. Coifman and Y. Meyer, Remarques sur l'analyse de Fourier á fenêtre, C. R. Acad. Sci. Paris, 312(1991), 259-261.

8. J. M. Combes, A. Grossman and P. Tchamitchian eds., Wavelets: Time-frequency methods and phase space, Springer-Verlag, Berlin, 1990.

9. W. Dahmen and C. Micchelli, On the optimal approximation rates for criss-cross finite element spaces, J. Comput. Appl. Math., 10(1984), 255-273.

10. W. Dahmen and C. Micchelli, Using the refinement equation for evaluating integrals of wavelets, SIAM J. Numer. Anal., 2(30), 1993, 507-537.

11. I. Daubechies, Ten lectures on wavelets, SIAM, Philadelphia, 1992.

12. G. Deslauriers and S. Dubuc, Interpolation dyadique, Fractals, dimensions non entierers et applications, G. Cherbit eds., Masson, Paris, 1987, 44-55.

13. S. Dubuc, Interpolation through an iterative scheme. J. Math. Anal. Appl., 114(1986), 185-204.

14. M. Farge, Wavelet transforms and their applications to turbulence, Ann. Rev. Fluid Mech., 24(1992), 395-457.

15. A. Grossman, R. Kronland-Martinet and J. Morlet, Reading and understanding continuous wavelet transforms, Wavelets: Time-frequency methods and phase space, J. M. Combes et al. eds., Springer-Verlag, Berlin, 1990, 2-20.

16. A. Grossman and J. Morlet, Decomposition of Hardy functions into square integrable wavelets of constant shape, SIAM J. Math., 15(1984), 723-736.

17. P. Dutilleux, An implementation of the "algorithme à trous" to compute the wavelet transform, Wavelets: Time-frequency methods and phase space, J. M. Combes et al. eds., Springer-Verlag, Berlin, 1990, 298-304.

18. C. E. Heil and D. F. Walnut, Continuous and discrete wavelet transforms, SIAM Review, 31(1989), 628-666.

19. M. Holschneider, R. Kronland-Martinet, J. Morlet and P. Tchamitchian, A real-time algorithm for signal analysis with the help of wavelet transform, Wavelets: Time-frequency methods and phase space, J. M. Combes et al. eds., Springer-Verlag, Berlin, 1990, 286-297.

20. R. Q. Jia and J. Lei, Approximation by multiinteger translates of functions having global support, JAT, 72(1993), 2-23.

21. W. Lawton, S. L. Lee and Z. Shen, Characterization of compactly supported refinable splines, manuscript.

22. S. Mallat and W. L. Hwang, Singularity detection and processing with wavelets, Robotics Report No. 245, Courant Institute of Mathematical Sciences, New York, NY, 1990.

23. S. Mallat and S. Zhong, Wavelet transform maxima and multiscale edges, Wavelets and their applications, G. Beylkin etl. eds., Jones & Bartlett, Boston, 1991, 147-171.

24. Y. Meyer, Ondelettes, Hermann, New York, 1988.

25. Y. Meyer, Wavelets, algorithms & applications, SIAM, Philadelphia, PA, 1993.

26. A. Oppenheim and R. W. Schafer, Discrete-time signal processing, Prentice Hall, Englewood Cliffs, New Jersey, 1989.

27. I. Schoenberg, Cardinal spline interpolation, SIAM, Philadelphia, PA, 1973.

28. G. Strang and [illegible] A Fourier analysis of the finite element variational method, Constructive a[illegible] of functional analysis, G. Geymonat Eds., C.I.M.E. Summer School, Cremonese, Rome, 1971, 793-840.

The results of this chapter are joint work with S. Riemenschneider.

# Chapter 3

## Image Compression By Wavelet Transform Coding

### §3.0. Introduction

Analyzing certain categories of signals is mathematically equivalent to analyzing certain groups of functions which belong to certain spaces of functions. To study these functions, we usually represent the function in a different form, e.g., the series expansion corresponding to a basis. In this chapter, we view series expansion as a transform process. With invertible transforms, the transform will theoretically represent all the information of the original signal. In some real applications, there may be certain advantages to code the transform rather than the original signal. This gives us a specific coding scheme: transform coding. We need to do some computation to obtain the transform and to get the signal back. Therefore, efficient computation schemes are needed for real computations. Normally bases which possess fast transforms are applied to this scheme.

For given signals, if we pick the right basis, the series expansion can be very sparse with most of the coefficients nearly equal to zero. Again with the proper choice of basis, these small coefficients contribute little to the original function. As an approximation, we may ignore the small coefficients by just setting them to be zero, thus keeping only the large coefficients. We may still get a good representation of the original signals even if we omit a lot of the small coefficients. Then by coding the remaining nonzero coefficients, we get what is called a compression scheme. Under the inverse transform, the original signal will not be obtained exactly in theory (unless the approximation is exact). Therefore this gives a lossy compression scheme, i.e.

some information is lost. However, in real applications, compression schemes can be extremely efficient, particularly when the compression is carried out by wavelet transforms [4][5][9].

Theoretically this compression scheme can be applied to all kinds of bases. Since compression is considered, we must represent the signal in a very efficient way without correlation. It is usually most efficient to use some orthogonal basis. We only study the transform coding compression scheme under orthogonal bases in this chapter. To simplify the discussion, the function spaces considered in this chapter are square integrable functions, namely $L_2(R)$ and $L_2(A)$, where $A$ is a finite interval. We denote both of them by $L_2(K)$, with $K$ denoting either the real line or a finite interval. All the bases used are assumed to be orthonormal in $L_2(K)$.

The efficiency of this scheme obviously depends on both the signal and the basis, i.e. it depends on the sparsity of the series expansion. As we will see from the discussion in the following sections, the efficiency will also depend on the sampling frequency of the signal since digitization is always involved in discrete signal processing. Several aspects of this scheme are discussed here with support of theoretical discussion and experimental results. All of our discussion is limited to one dimensional functions since the tensor product approach can be used to deal with image compression.

## §3.1. Compression Through Transform Coding

Decomposition: Let $f \in L_2(K)$, $W = \{w_i\}_{i \geq 1}$ be an orthonormal basis of $L_2(K)$, and $f = \sum_{i \geq 1} a_i w_i$ be the orthogonal expansion of $f$ with respect to the basis $W$. Then define the decomposition transform by

$$D(W, \cdot) : L_2(K) \longrightarrow l_2(Z) : f \mapsto \{a_i\}_{i \geq 1}.$$

A partial decomposition transform can be defined by

$$D_n(W,\cdot) : L_2(K) \longrightarrow l_2(Z) : f \mapsto \{a_i\}_{1\leq i\leq n}.$$

Reconstruction: Let $A = \{a_i\}_{i\geq 1} \in l_2(Z)$, and $W = \{w(t)\}_{i\geq 1}$ be an orthonormal basis of $L_2(K)$. Then the reconstruction transform is defined by

$$R(W,\cdot) : l_2(Z) \longrightarrow L_2(K) : \{a_i\}_{i\geq 1} \mapsto \sum_{i\geq 1} a_i w_i.$$

Likewise, a partial reconstruction transform is defined by:

$$R_n(W,\cdot) : l_2(Z) \longrightarrow L_2(K) : \{a_i\}_{1\leq i\leq n} \mapsto \sum_{1\leq i\leq n} a_i w_i.$$

In this setting, the coefficients $A = \{a_i\}_{i\geq 1}$ gives all the information about $f$. Mathematically, it is equivalent to study either the coefficients or the signal $f$ itself, i.e. properties of $f$ are found by studying the coefficients and vice versa. Here we cite only the following energy preservation equation which is known as Parseval's identity.

**Theorem 1.1.** *If $f \in L_2(K)$ and $W = \{w_i\}_{i\geq 1}$ is an orthonormal basis of $L_2(K)$, then*

$$||f||_2 = ||D(W,f)||_2.$$

*The quantity $||f||_2^2 = \int_R f^2(t)dt$ is the* ***energy*** *of $f$, while the quantity $||A||_2^2 = \sum_{i\geq 1} a_i^2$ is the* ***energy*** *of the sequence $A$.*

**Lemma 1.2.** *If $W = \{w_i\}_{i\geq 1}$ is an orthonormal basis of $L_2(K)$, then*

*1.* $||D(W,\cdot)|| = ||D_n(W,\cdot)|| = 1.$

*2.* $\lim_{n\to\infty} ||D_n(W,\cdot) - D(W,\cdot)|| = 0.$

*3.* $||R(W,\cdot)|| = ||R_n(W,\cdot)|| = 1.$

*4.* $\lim_{n\to\infty}||R_n(W,\cdot) - R(W,\cdot)|| = 0.$

*5.* $R(W,\cdot)D(W,\cdot) = I$*, the identity on* $L_2(K)$.

*6.* $R(W,\cdot)D(W,\cdot) = I$*, the identity on* $l_2(Z)$.

Proof: (1) and (3) follows from the Parseval identity and the facts $D_n(W, w_i) = e_i$, $||R(W, e_i)|| = w_i$, if $i \leq n$, where $e_i(j) = \delta_{ij}$. (2), (4), (5) and (6) follow from the definitions of decomposition and reconstruction. □

If we code $A = \{a_i\}_{i\geq 1}$ instead of coding the original signal $f$ directly, we get the transform coding scheme. Theoretically, the transform coding technique can be applied associate even with nonorthogonal bases as long as we can have invertible transform. For real time applications, we need computational speed when applying the transform coding scheme. That is, we need to pick the right basis so that there is fast decomposition and fast reconstruction. The orthogonality requirement is to obtain a more efficient representation for the compression.

For nice bases, the computation can be very efficient. It is in these cases that there are quite a few fast transforms, for example the traditional FFT/IFFT, the newly developed FWT/IFWT, and the lapped transform for the Malvar basis. In real applications, the signal is normally in discrete form and the data length is finite. Therefore, partial decomposition and partial reconstruction are actually applied.

Now let's introduce the compression scheme through transform coding. Before we code $A = \{a_i\}_{i\geq 1}$, we do some approximation to $A$, $A^* = \{a_i^*\}_{i\geq 1}$, and code $A^*$ instead. If $A^*$ takes less storage after coding, then we gain a compression scheme:

$$C : l_2(Z) \longrightarrow l_2(Z),\ C(A) = A^*.$$

A partial compression scheme can be similarly described:

$$C_n(\{a_i\}_{1\le i\le n}) = \{a_i^*\}_{1\le i\le n}.$$

This just describes the process, the definition for the exact implementation scheme will depend on the particular application.

To get the original signal back, we need to reconstruct the compressed signal $f^* = R(W, A^*)$. Normally, $f \neq f^*$ unless $A = A^*$. Therefore, the complete process of the compression scheme can be described as follows.

- Digitize the original signal $f$ to get a finite length discrete version.
- Partially decompose the discrete version (fast transform of discrete signal).
- Compression via coding of the coefficients of the decomposition.
- Storage or transfer.
- Decode and reconstruction (fast inverse transform).

Due to storage limitations and the transfer speed, compression is often involved in some applications. Several different compression schemes have been studied and applied. These schemes can be divided into two categories, **lossless** and **lossy** compression. The difference is whether after uncompression we can get a signal identical to the original one or not. Compression through transform coding is clearly a lossy scheme. Lossy compression certainly introduces errors and therefore, the error must be well controlled in real applications.

**Theorem 1.3.** *If $f \in L_2(K)$ and $W = \{w_i\}_{i\ge 1}$ is an orthonormal basis of $L_2(K)$, then*

$$||f - R(W, CD(W, f))||_2 \le ||I - C|| \cdot ||f||_2.$$

Proof: By the Parseval identity, we get

$$||f - R(W, CD(W,f))||_2 = ||D(W,f) - CD(W,f)||_2.$$

From the defini ion of $C$, we have

$$\begin{aligned} & ||D(W,f) - CD(W,f)||_2 \\ = & ||(I - C)D(W,f)||_2 \\ \leq & ||I - C|| \cdot ||D(W,f)||_2 \\ = & ||I - C|| \cdot ||f||_2 \end{aligned}$$

which is the result we want. □

**Theorem 1.4.** *If $f \in L_2(K)$ and $W = \{w_i\}_{i\geq 1}$ is an orthonormal basis of $L_2(K)$, then*

$$||f - R_n(W, C_n D_n(W,f))||_2 \leq (||I_n - C_n|| + ||D_n - D||)||f||_2$$

*where $I_n(\{a_i\}_{1\leq i\leq n}) = \{a_i\}_{1\leq i\leq n}$ is an identity.*

Proof: By the definitions of decomposition and reconstruction, we have

$$\begin{aligned} f & = R(W, D(W,f)) \\ & = R(W, D_n(W,f) + (D - D_n)(W,f)) \\ & = R(W, D_n(W,f)) + R(W, (D - D_n)(W,f)) \\ & = R_n(W, D_n(W,f)) + R(W, (D - D_n)(W,f)). \end{aligned}$$

Therefore, we have

$$\begin{aligned} & ||f - R_n(W, C_n D_n(W,f))||_2 \\ = & ||R_n(W, D_n(W,f)) + R(W, (D - D_n)(W,f)) - R_n(W, C_n D_n(W,f))||_2, \end{aligned}$$

and some standard estimates will give the result we want. □

The above simple results tell us that compression error is mainly determined by $||I_n - C_n||$ which depends on the compression scheme we use, and by $||D_n - D||$ which depends on the approximation power of the subspace $V_n = span\{w_i\}_{1\leq i\leq n}$. For efficiency, we normally apply an orthogonal basis which possesses good approximation power.

## §3.2. Two Basic Compression Schemes

For given signal and basis, the compression error is mainly determined by the approximation error introduced by the compression. More precisely, in order to reduce the compression error, we need to reduce

$$||I - C||$$

or

$$||I_n - C_n||$$

in the case of real implementation. If we consider just the error, the best choice in theory is given by: $C = I$ and $C_n = I_n$; the case without any approximation. Unfortunately, this leads to no compression at all. Therefore, we have to allow compression error to exist in some acceptable range and design the compression scheme accordingly.

Another index to measure the efficiency of the compression scheme is the compression ratio (we also call it *compression rate*, or simply *rate*). Intuitively, compression ratio is for measuring the success of compression. Notice that depending on different compression schemes, the compression ratio can be defined in several different ways. For example, we can define the compression ratio as the size ratio between the file which stores the compressed data and the file which stores the original data. Before we introduce a definition of compression ratio, we first introduce the following two basic compression schemes:

- Scheme I: Given $0 < \varepsilon < 1$ and $\{a_i\}_{1\leq\cdot\leq n}$, let $p$ be the largest integer such that $p \leq n\varepsilon$. A set $\mathcal{J}$ of cardinality $p$ is chosen among the coefficients with the following property: if $j \in \mathcal{J}$, then

$$|a_j| \geq |a_i|,\ i \notin \mathcal{J}.$$

Then define

$$a_i^* = \begin{cases} 0, & i \notin \mathcal{J}, \\ a_i, & i \in \mathcal{J}. \end{cases}$$

The compression scheme $C_n$ is defined by: $C_n(\{a_i\}_{1\leq i\leq n}) = \{a_i^*\}_{1\leq i\leq n}$.

- Scheme II: For $\delta > 0$, $C_n(\{a_i\}_{1\leq i\leq n}) = \{a_i^*\}_{1\leq i\leq n}$, where $a_i^*$, $1 \leq i \leq n$, is defined by:

$$a_i^* = \begin{cases} 0, & |a_i| \leq \delta, \\ a_i, & \text{else}. \end{cases}$$

Here $\delta$ is called the **tolerance**.

In either of the above compression schemes, we have: $a_i^* = a_i$, or $a_i^* = 0$ depending only on the magnitude of $a_i$. Therefore, we define the compression ratio as:

$$\rho = \tfrac{1}{n}\textstyle\sum_{1\leq i\leq n, a_i^*\neq 0} 1.$$

Clearly, the $\varepsilon$ given in Scheme I is essentially the compression ratio and is fixed, while the compression ratio in Scheme II strongly depends on the data.

**Theorem 2.1.** *Suppose $\rho$ is the given compression ratio under Scheme I. Then*

$$||f - R_n(W, C_n D_n(W, f))||_2 \leq (\sqrt{1-\rho} + ||D - D_n||)||f||_2.$$

Proof: Suppose $D_n(W,f) = \{a_i\}_{1\leq i\leq n}$ and $C_n(\{a_i\}_{1\leq i\leq n}) = \{a_i^*\}_{1\leq i\leq n}$. Scheme I gives $a_i^* = a_i$, or $a_i = 0$. Therefore,

$$\frac{\sum_{i\in\mathcal{J}}|a_i|^2}{\sum_{1\leq i\leq n}|a_i|^2} \geq \rho.$$

Therefore, we have

$$\sum_{1\leq i\leq n}|a_i - a_i^*|_2^2 \leq (1-\rho)\sum_{1\leq i\leq n}|a_i|_2^2,$$

which implies

$$||I_n - C_n|| \leq \sqrt{1-\rho}.$$

This gives the result with the help of theorem 1.2. □

**Theorem 2.2.** *Suppose $\delta$ is the given tolerance under Scheme II, then*

$$||f - R_n(W, C_n D_n(W,f))||_2 \leq (\sqrt{n}\delta + ||D - D_n||)||f||_2.$$

Proof: Suppose $D_n(W,f) = \{a_i\}_{1\leq i\leq n}$ and $C_n(\{a_i\}_{1\leq i\leq n}) = \{a_i^*\}_{1\leq i\leq n}$, then Scheme II gives us

$$|a_i - a_i^*| < \delta.$$

Therefore $\sum_{1\leq i\leq n}|a_i - a_i^*|^2 \leq n\delta^2$, which implies

$$||I_n - C_n|| \leq \sqrt{n}\delta$$

and this completes our proof with the help of Theorem 1.2. □

The main advantage of the first approach is that the compression is according to a previously given compression ratio. The disadvantage is that a selection procedure to sort out the large coefficients is required and this is typically done by a sorting algorithm which will slow down the computation speed. Another problem is that without knowledge of the distribution of the coefficients it will be hard to control the error. For example, if all the coefficients are almost equal in magnitude and we set 0.05 as the compression ratio, the error will be extremely large (the relative error roughly equals 0.95). The advantage of the second approach is that through the choice of tolerance we have some control of the compression error. But the compression ratio will fluctuate depending on the signal, the basis and the tolerance.

Since partial decomposition and reconstruction are always involved in real applications, a natural question is the behavior of these compression schemes for a large $n$. Under the given conditions, we have the following asymptotic results about the compression.

**Theorem 2.3.** *Suppose $\rho$ is the given compression ratio under Scheme I. Then*

$$\lim_{n\to\infty}||f - R_n(W, C_n D_n(W, f))||_2 = 0.$$

Proof: Let $f = \sum_{1\leq i\leq\infty} a_i w_i$. For any $\lambda > 0$, there exist $K$ such that $N \geq K$ implies $||f - g_N||_2 < \lambda$, where $g_N = \sum_{1\leq i\leq N} a_i w_i$. Let $g_N^* = R_N(W, C_N D_N(W, f))$. According to the first compression scheme, for $N$ large enough, the cardinality of the set $\mathcal{J}$ is bigger than $K$; hence $||f - g_N^*||_2 \leq \lambda$. □

**Theorem 2.4.** *Let $\lambda > 0$, $f \in L_2(R)$. There exist $N_0 > 0$ and $\varepsilon > 0$, such that under Scheme II with $n > N_0$ and tolerance $\delta \leq \varepsilon$, we have*

$$||f - R_n(W, C_n D_n(W, f))||_2 \leq \lambda.$$

Proof: Let $f = \sum_{1\leq i\leq\infty} a_i w_i$. For any $\lambda > 0$, there exist $K$ such that $N \geq K$ implies $||f - g_N||_2 < \lambda$, where $g_N = \sum_{1\leq i\leq N} a_i w_i$. Let $\varepsilon = \min\{|a_i| : a_i \neq 0,\ 1 \leq i \leq K\}$, $N_0 = K$ and take $g_N^* = R_N(W, C_N D_N(W, f))$. According to Scheme II, for $N > N_0$ and $\delta \leq \varepsilon$, we have $||f - f_N^*||_2 \leq ||f - g_N||_2 < \lambda$. □

From these results, we can see that if we keep increasing $n$ in the partial decomposition, the compression quality can be guaranteed by both of these schemes. In other words, as the sampling frequency increases, we always end up with good compression (lower compression error and lower compression ratio). However, the dilemma is that increasing the sampling frequency means more data while the purpose of compression is to carry less data. Another problem is that to keep increasing the sampling frequency is not always practical.

Clearly the compression ratio is determined by both the signal and the basis used in the compression scheme. More precisely, it is determined by the sparsity of the series expansion. Therefore, basis functions which better *mimic* the shape of the signal are normally more efficient. If the signal is just a finite linear combination of the basis functions, then obviously the best basis has been found. Clearly, no best basis for a compression scheme can be found without a priori knowledge about the signal. A more efficient implementation of this scheme comes from the problem criteria and knowledge of the signals.

**Example 2.5.** If we use the Fourier basis and

$$f(t) = \begin{cases} -1, & -1 \leq t < 0, \\ 1, & 0 < t \leq 1, \end{cases}$$

then we have

$$f(t) = \tfrac{4}{\pi}\textstyle\sum_{n\geq 0}\frac{\sin(2n+1)\pi t}{2n+1}.$$

However, if we apply Haar basis and let $\psi(t)$ be the Haar wavelets, then

$$f(t) = \psi(2^{-1}t + 1).$$

Therefore the compression by the Haar basis is far more efficient. This example also shows us that if the signal is not smooth enough, applying the compression scheme with smoother basis does not necessarily give better results.

**Example 2.6.** If $f(t) = \chi_{[-1,1]}\sin \pi t$ and we use the Fourier basis, then

$$f(t) = \sin \pi t, \quad t \in [-1,1]$$

where the Haar series has

$$< f, \psi_{i,j} >= \tfrac{1}{\pi}[2\cos \tfrac{2j+1}{2^{i+1}}\pi - \cos \tfrac{j}{2^i}\pi - \cos \tfrac{j+1}{2^i}\pi], \quad \text{for } -2^i < j < 2^i$$

Therefore we have an infinite wavelet series and the compression by Fourier basis is much better in this case.

## §3.3. Wavelet Transforms

A sequence of subspaces $V_k \subseteq L_2(R)$ forms a **multiresolution analysis** of $L_2(R)$ [18], if they satisfy the following conditions:

1. $\cdots \subseteq V_{-1} \subseteq V_0 \subseteq V_1 \subseteq \cdots$.
2. $\bigcup_{i\in Z} V_i$ is dense in $L_2(R)$.
3. $\bigcap_{i\in Z} V_i = \{0\}$.
4. $\phi(\cdot) \in V_k$ if and only if $\phi(2\cdot) \in V_{k+1}$.
5. There exist Riesz basis for $V_0$. i.e. there exists $\phi \in V_0$ such that $\{\phi(\cdot - j)\}_{j\in Z}$ forms a basis for $V_0$. Also $\varphi$ satisfies $C_1||a||_2 \leq ||\Sigma_{j\in Z} a_j \phi(\cdot - j)||_2 \leq C_2||a||_2$, for all $a(\cdot) \in l_2$, with fixed constants $C_1$ and $C_2$ such that $0 < C_1 < C_2$.

If we write $V_{i+1} = V_i \oplus W_i$ as an orthogonal direct sum, then $W_i \perp W_j$ if $i \neq j$ and $\oplus_{i\in Z} W_i$ is dense in $L_2(R)$. Denote $\psi_{i,j} = \psi(2^i \cdot - j)$, for $i, j \in Z$. If there exists a function $\psi$ such that $\psi(\cdot - j) \perp \psi(\cdot - l)$, $j \neq l$ and

$$W_i = span\{\psi_{i,j} : j \in Z\}, \quad i \in Z,$$

then $\psi_{i,j} \perp \psi_{k,l}$ for $i \neq k$ or $j \neq l$, and $\psi$ is called an orthogonal wavelet.

**Theorem 3.1.** [8] *If $\{V_k\}$ forms a multiresolution analysis of $L_2(R)$, then there exists $\phi$ such that $\{\phi(\cdot - j)\}_{j \in Z}$ forms an orthonormal basis of $V_0$, and satisfying $\phi = \sum_{k \in Z} h(k)\phi(2 \cdot -k)$, where*

$$\psi = \sum_{k \in Z}(-1)^k h(k-1)\phi(2 \cdot -k)$$

*is an orthogonal wavelet of $L_2(R)$.*

The sequence $\{h(k)\}_{k \in Z}$ is called the **mask sequence** and the sequence $\{g(k)\}_{k \in Z} = \{(-1)^k h(k-1)\}_{k \in Z}$ is called the **auxiliary sequence**. All the properties of $\phi$, and thus of the wavelet $\psi$, are fully represented by the mask sequence [18]. Normally we require a wavelet with fast decay in both time and frequency domain in order to gain **local analysis**. This **locality** is usually expressed by exponential decay or compact support (there cannot be compact support in both time and frequency domain). It is possible to construct a mask sequence of finite length which gives rise to compactly supported orthogonal wavelets (but a finitely supported mask sequence does not guarantee this). The compactly supported wavelets can have arbitrarily high regularity at the expense of a longer mask sequence, thus a longer support interval for the wavelet. The Daubechies family of wavelets are excellent examples of compactly supported orthogonal wavelets [6]. Our image compression examples in the last section are done by using Daubechies' wavelets $D_2$, $D_4$ and $D_6$. $D_2$ and the well known Haar basis.

Wavelets provide us with a way to decompose functions into multichannels according to the frequency or in other words resolution. With this decomposition, we gain very efficient function representation and the signal can be filtered according to different frequencies. The following diagram shows this decomposition:

$$\begin{tikzcd}
V_{i+k} \arrow[r] \arrow[dr] & V_{i+k-1} \arrow[r] \arrow[dr] & \cdots & V_i \\
& W_{i+k-1} & \cdots & W_i
\end{tikzcd}$$

For a function $f \in L_2(R)$, define the following projection operators

$$P_i : \quad f \longmapsto f|_{V_i}$$

$$Q_i : \quad f \longmapsto f|_{W_i}.$$

Then the above decomposition can be expressed in the following way:

$$\begin{tikzcd}
P_{i+k}f \arrow[r] \arrow[dr] & P_{i+k-1}f \arrow[r] \arrow[dr] & \cdots & P_i f \\
& Q_{i+k-1}f & \cdots & Q_i f
\end{tikzcd}$$

Multiresolution analysis and orthogonal wavelets provide a very efficient algorithm known as the FWT (fast wavelet transform) or Mallat's algorithm. FWT refers to two different procedures in the multiresolution analysis: decomposition and reconstruction. Sometimes the two procedures are distinguished with FWT referring to decomposition while reconstruction is referred to as IFWT (inverse fast wavelet transform). Let $\{a(i,j)\}_{j\in Z} = \{2^{i/2} < f, \phi_{i,j} >\}_{j\in Z}$ and $\{b(i,j)\}_{j\in Z} = \{2^{i/2} < f, \psi_{i,j} >\}_{j\in Z}$. Then from the definitions of the mask sequence and the auxiliary sequence, the coefficients from level $i+1$ (scale $2^{i+1}$) to level $i$ (scale $2^i$) are related by

$$a(i,j) = \frac{1}{\sqrt{2}} \sum\nolimits_{n\in Z} h(n-2j)a(i+1,n),$$

$$b(i,j) = \frac{1}{\sqrt{2}} \sum\nolimits_{n\in Z} g(n-2j)a(i+1,n).$$

Thus in terms of coefficient sequences, decomposition is described schematically by

$$\begin{tikzcd}
a(i+k,\cdot) \arrow[r] \arrow[dr] & a(i+k-1,\cdot) \arrow[r] \arrow[dr] & \cdots & a(i,\cdot) \\
& b(i+k-1,\cdot) & \cdots & b(i,\cdot)
\end{tikzcd}$$

Likewise, the coefficients of level $i+1$ are reconstructed from level $i$ by

$$a(i+1,j) = \frac{1}{\sqrt{2}}\sum\nolimits_{n\in Z} h(j-2n)a(i,n) + \frac{1}{\sqrt{2}}\sum\nolimits_{n\in Z} g(j-2n)b(i,n);$$

Schematically by,

$$\begin{tikzcd}
a(i,\cdot) \arrow[r] & a(i+1,\cdot) \arrow[r] & & \cdots \; a(i+k,\cdot) \\
b(i,\cdot) \arrow[ur] & \cdots & b(i+k-1,\cdot) \arrow[ur] &
\end{tikzcd}$$

The most remarkable property of wavelets is probably their locality which is extremely efficient in dealing with singularities [15][16]. More precisely, the scaling/shift and locality combined together allows wavelet decomposition to mimic the shape of a given signal normally better than with other bases. Consequently, wavelet expansions in the above schemes are normally very sparse, i.e. the magnitude of most of the coefficients is close to zero. Therefore wavelet transform coding is very good for compression. These nice properties combined with the efficiency of FWT makes compression by wavelet transform coding successful in many real applications [4][5][9].

Wavelet analysis gives the decomposition $V_{i+1} = V_i \oplus W_i$. For a finer resolution decomposition, multiresolution analysis has been generalized to wavelet packet analysis [19]. Wavelet packet analysis gives an orthogonal decomposition of subspace $W_n$. If we denote

$$w_{2n} = \sum h(k)w_n(2\cdot -k),$$

$$w_{2n+1} = \sum g(k)w_n(2\cdot -k)$$

with $w_0 = \phi$ and $w_1 = \psi$. Under the conditions of multiresolution, we have [19]

**Theorem 3.2.** *Under the given conditions, the functions*

$$\{2^{i/2}w_n(2^i\cdot -j)\}_{i,j\in Z}$$

*with $2^l \leq n < 2^{l+1}$ for fixed $l \geq 0$ form an orthonormal basis of $L_2(R)$.*

Let $V_{n,i} = span\{w_n(2^i \cdot -j)\}_{j\in Z}$, then the wavelet packet analysis can be described schematically as follows:

$$\begin{tikzcd}
 & & V_{4n,i-2} \\
 & V_{2n,i-1} \arrow[ur] \arrow[dr] & \\
 & & V_{4n-1,i-2} \\
V_{n,i} \arrow[uur] \arrow[ddr] & & \\
 & & V_{4n-2,i-2} \\
 & V_{2n-1,i-1} \arrow[ur] \arrow[dr] & \\
 & & V_{4n-3,i-2}
\end{tikzcd}$$

This makes multiresolution a special case of wavelet packet analysis. If $\{a(n,i,j)\}_{j\in Z} = \{2^{i/2} < f,(w_n)_{i,j} >\}_{j\in Z}$, then decomposition and reconstruction in wavelet packet analysis can be described by the following Split Algorithm:

Decomposition: $a(2n,i,j) = \frac{1}{\sqrt{2}}\sum_{k\in Z}h(k-2j)a(n,i+1,k)$,

$$a(2n-1,i,j) = \frac{1}{\sqrt{2}}\sum_{k\in Z}g(k-2j)a(n,i+1,k).$$

$$\begin{tikzcd}
 & & a(4n,i-2,\cdot) \\
 & a(2n,i-1,\cdot) \arrow[ur] \arrow[dr] & \\
 & & a(4n-1,i-2,\cdot) \\
a(n,i,\cdot) \arrow[uur] \arrow[ddr] & & \\
 & & a(4n-2,i-2,\cdot) \\
 & a(2n-1,i-1,\cdot) \arrow[ur] \arrow[dr] & \\
 & & a(4n-3,i-2,\cdot)
\end{tikzcd}$$

Reconstruction: $a(n,i+1,j)=\frac{1}{\sqrt{2}}\sum_{k\in Z}h(j-2k)a(2n,i,k)$

$$+\frac{1}{\sqrt{2}}\sum_{k\in Z}g(j-2k)a(2n-1,i,k).$$

$$\begin{tikzcd}
a(4n,i-2,\cdot) \arrow[dr] & & \\
 & a(2n,i-1,\cdot) \arrow[ddr] & \\
a(4n-1,i-2,\cdot) \arrow[ur] & & \\
 & & a(n,i,\cdot) \\
a(4n-2,i-2,\cdot) \arrow[dr] & & \\
 & a(2n-1,i-1,\cdot) \arrow[uur] & \\
a(4n-3,i-2,\cdot) \arrow[ur] & &
\end{tikzcd}$$

From an algorithmic point of view, the Split Algorithm is just applying FWT in a balanced tree manner. Therefore the numerical performance of this algorithm is the same as the FWT. Actually before wavelet analysis, the signal processing community had designed a scheme called quadrature mirror filters and a fast algorithm called the Pyramid Algorithm [2]. From the algorithmic point of view, FWT and Split Algorithm can be viewed as special cases of this more general algorithm. However, there is multiresolution analysis behind the FWT and Split Algorithm while there is no such structure for general quadrature mirror filters. The significance of wavelet packet analysis is that the spaces involved in the multiresolution can be freely decomposed according to an adaptive best basis selection scheme with the help of a cost index such as entropy [4]. This turns out to be an extremely powerful compression scheme for images.

## §3.4. Image Compression

Mathematically, an image is a function of two variables defined on a compact (normally rectangular) domain described by values at a discrete set of points (called pixels). Therefore an image can be viewed as a function defined on certain grid or simply treated as a matrix. We only consider $f \in L_2(K \times J)$, with $K$, $J$ denote finite intervals on the real line. There are many discussions about intrinsically multidimensional orthogonal basis, i.e. the results about the construction of intrinsically multidimensional orthogonal wavelets [22]. In order to simplify our discussion, we only consider tensor product wavelets In that case we have the following well known result(cf [8][18][19]).

**Proposition 4.1.** *If $\{W_i\}_{i\geq 1}$ and $\{U_i\}_{i\geq 1}$ are orthonormal bases of $L_2(K)$ and $L_2(J)$ respectively, then $\{W_i(x)U_j(y)\}_{i,j\geq 1}$ is an orthonormal basis of $L_2(K \times J)$.*

Thus all the mathematics involved in tensor product case is virtually the single variable case. Therefore the analysis about signal compression can be generalized to image compression in this case. As far as real implementations are concerned, we can decompose each row then each column or vise versa. In the reconstruction, the inverse transform is performed on each column and then on each row, in reverse order to that of decomposition. From the algorithmic point of view, the main advantage of the tensor product approach is that it can be implemented in parallel which will be very efficient.

## §3.5. Examples and Experiments

**Example 5.1.** Multiresolution analysis gives us a way to decompose signals into multiple channels according to different frequencies. Here we present in Figures 1-2, the decomposition of a $512 \times 512 \times 8bpp$ (*bpp* is the abbreviation for *bits per pixel* ) picture of Jupiter. If Figure 1 represents $V_5 \times V_5$, then it is decomposed by tenser products of $V_0$ and $\hat{W}_0 = W_0 \oplus \cdots \oplus W_4$. the four figures in Figure 2 represent the projections $V_0 \times V_0$, $V_0 \times \hat{W}_0$, $\hat{W}_0 \times V_0$ and $\hat{W}_0 \times \hat{W}_0$ respectively. The decomposition is conducted to the fifth level by $D_4$ and the four components are corresponding to $\phi(x)\phi(y)$, $\phi(x)\psi(y)$, $\psi(x)\phi(y)$ and $\psi(x)\psi(y)$. This example illustrates the sparsity of the wavelet series expansion.

**Example 5.2.** Now let's look at the image compression by wavelet transforms. In order to avoid the edge effects, first we use the Haar basis to do the compression to an $480 \times 512 \times 8bpp$ image. We applied Scheme I to do this example and the results are listed in Figures 3-12. As we can observe from these experiments, we can see that the level of decomposition plays an important rule. Yet it does not necessarily mean deeper level decomposition will give better results. Another observation is that when the decomposition level is not deep enough, the energy error is relatively small even though the visual result is not good.

**Example 5.3.** Now let's look at the image compression with wavelet transforms by using $D_4$. This time we use Scheme II to repeat the previous example for an $480 \times 512 \times 8bpp$ image. We list the results in Figures 13-23. Similar to the fixed ratio compression case, the compression ratio and error are related to the level of decomposition. As showed by the pictures, as the level goes deeper, the compression ratio and error dropped down yet the visual quality of the image remains basically the same. Also we observe that the quality of the image is not fully represented by the energy loss.

**Example 5.4.** Now let's look at the image compression with wavelet transforms by using $D_4$. Scheme I is applied here to a $512 \times 512 \times 8bpp$ image. We list the results in

Figures 24-25. Since a sorting algorithm is needed to get the portion of coefficients with largest magnitude, the fixed ratio compression scheme is slower than the non-fix ratio compression. However, to speed up the computation, we apply the sorting algorithm only to each row and select coefficients based on a row by row basis. The results are listed in Figures 26-27. We call this scheme as pseudo compression and the given compression ratio is also called pseudo ratio. Clearly we see that the quality is not as good as the results obtained by global sorting. Yet the speed is much faster and can be implemented by parallel computation.

**Remark:** From the above examples, we can see that even a direct application of the wavelet transform can turn out some very nice results. These results once again demonstrate the power of wavelet analysis. From our discussions, we can see clearly that no best basis exists for compression in general. In image compression, the level of decomposition plays an important role. From this point of view, the wavelet packets technique is more adaptive with the help of entropy. Yet the problem with entropy is that we know entropy only after transform. And this limits its function as index of basis selection. Therefore the choices of the best basis selection or adaptive scheme remains an very interesting question both in theory and in application.

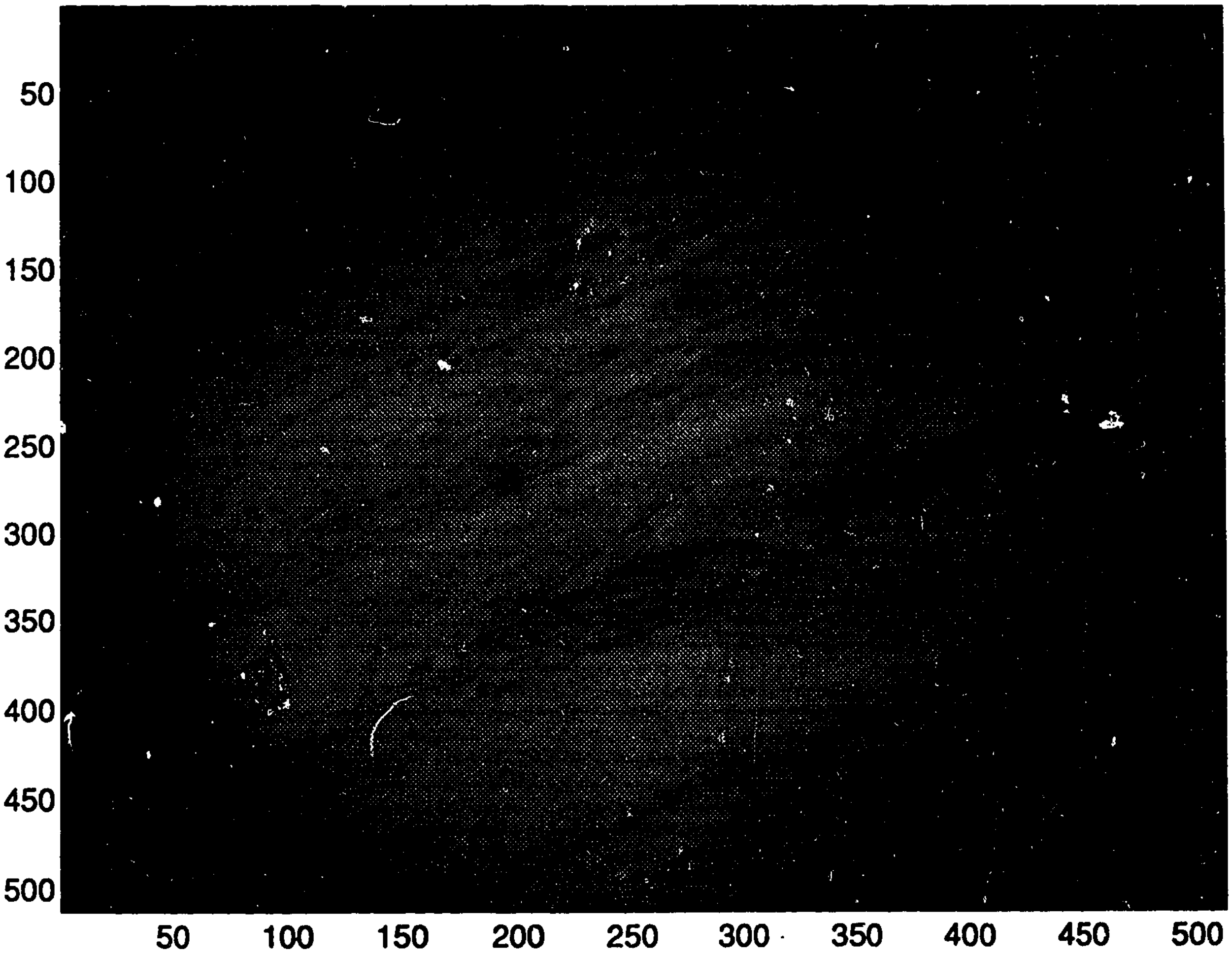


**Figure 1.** (Example 5.1) A 512 × 512 × 8*bpp* digital image of Jupiter.

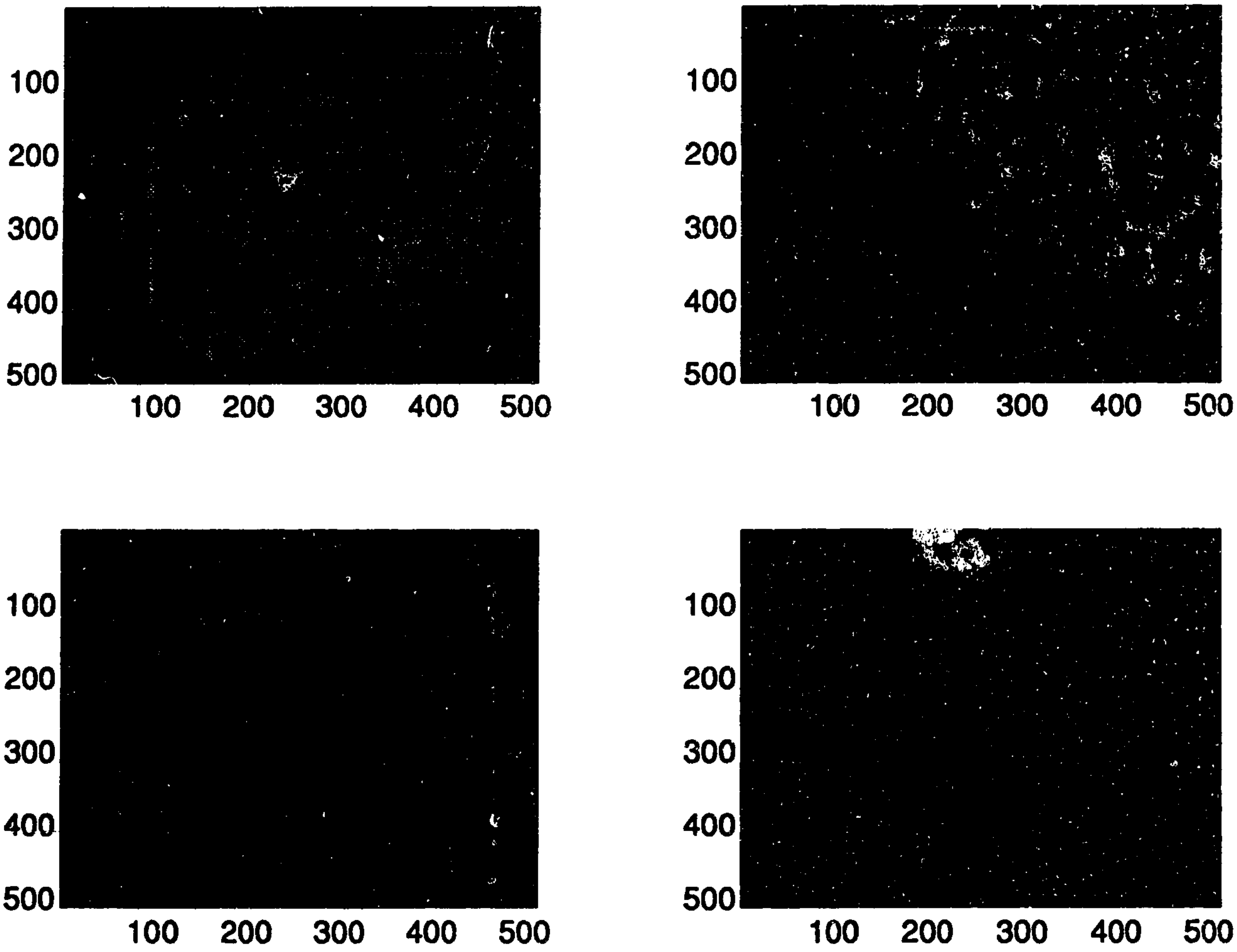


**Figure 2.** (Example 5.1) Four components of Figure 1 under 5 levels of decomposition using D4: $\phi(x)\phi(y)$, $\phi(x)\psi(y)$, $\psi(x)\phi(y)$, $\psi(x)\psi(y)$ parts with left-right top-bottom order.

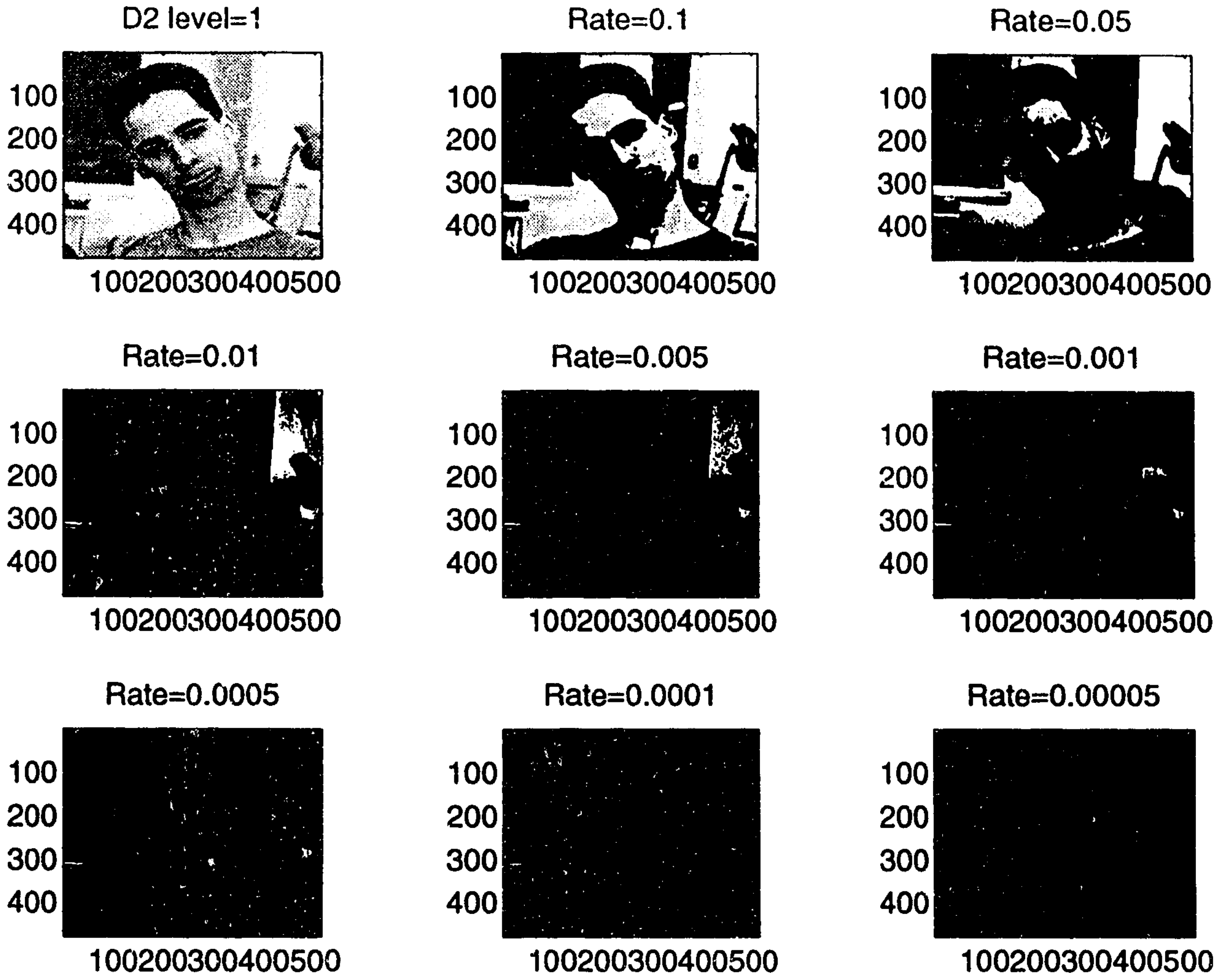


**Figure 3.** (Example 5.2) Scheme I compression with 1 level of decomposition using the D2 transform at various compression rates. The original image is at top left.

]

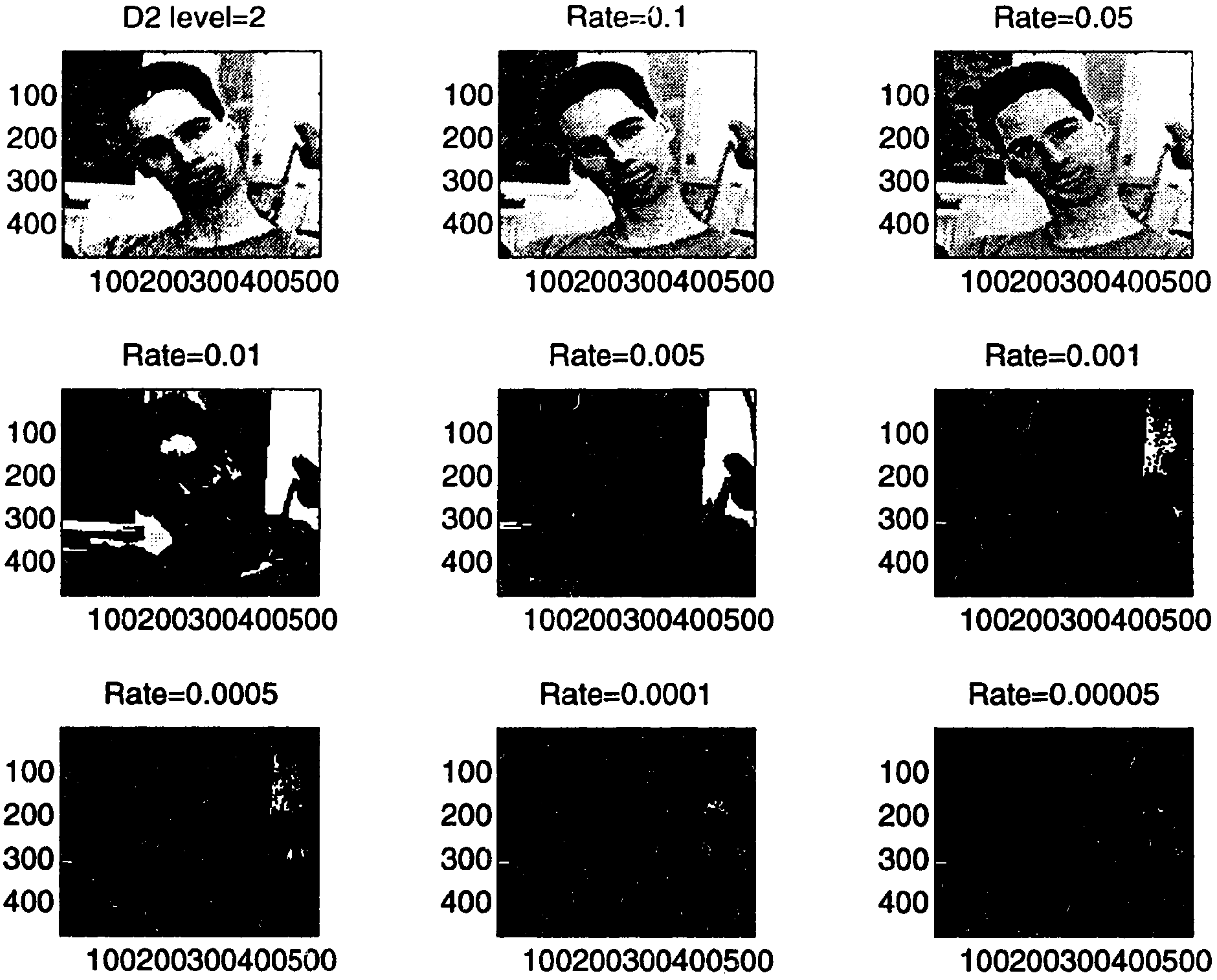


**Figure 4.** (Example 5.2 ) Scheme I compression with 2 levels of decomposition using the D2 transform at various compression rates. The original image is at top left.

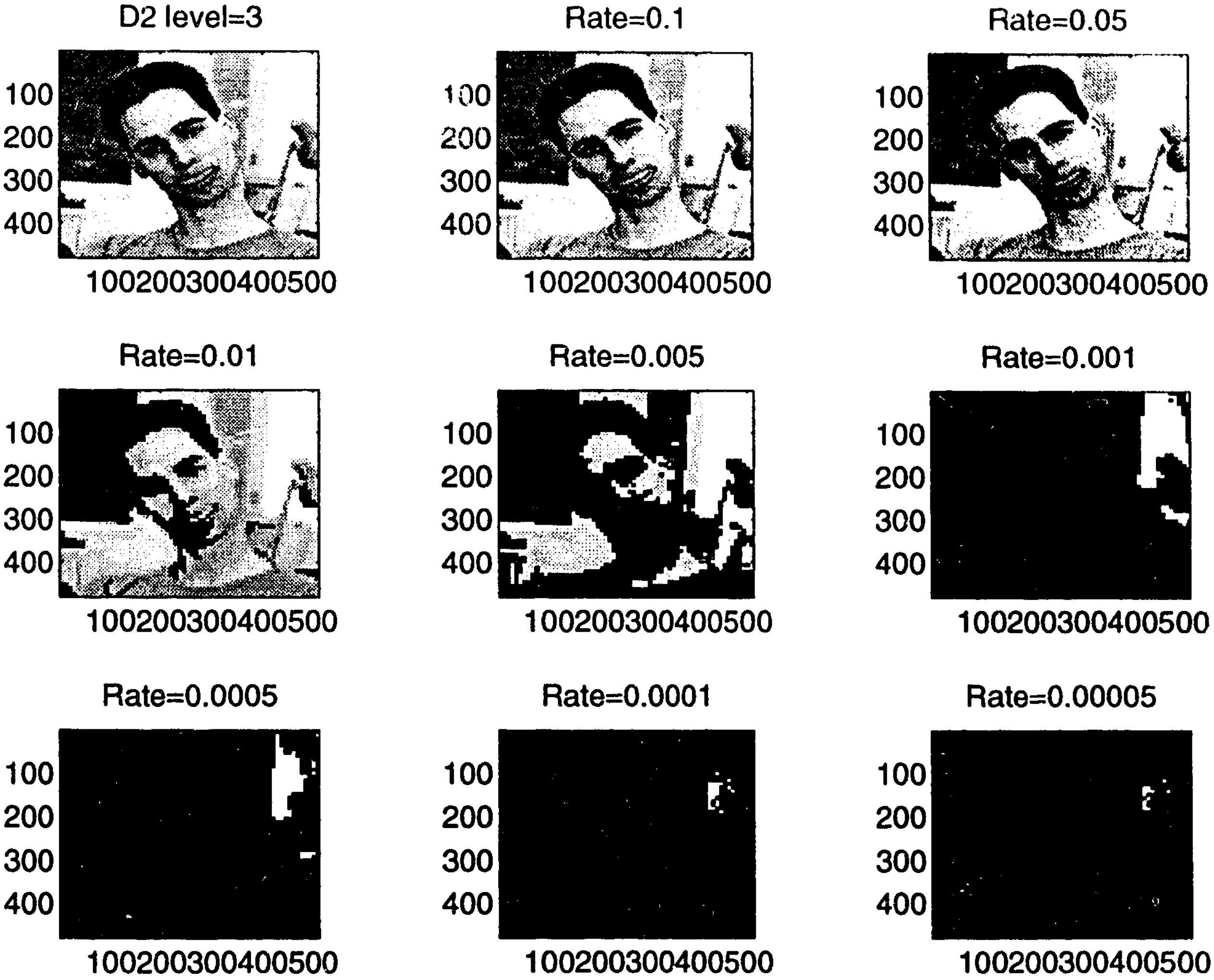


**Figure 5.** (Example 5.2 ) Scheme I compression with 3 levels of decomposition using the D2 transform at various compression rates. The original image is at top left.

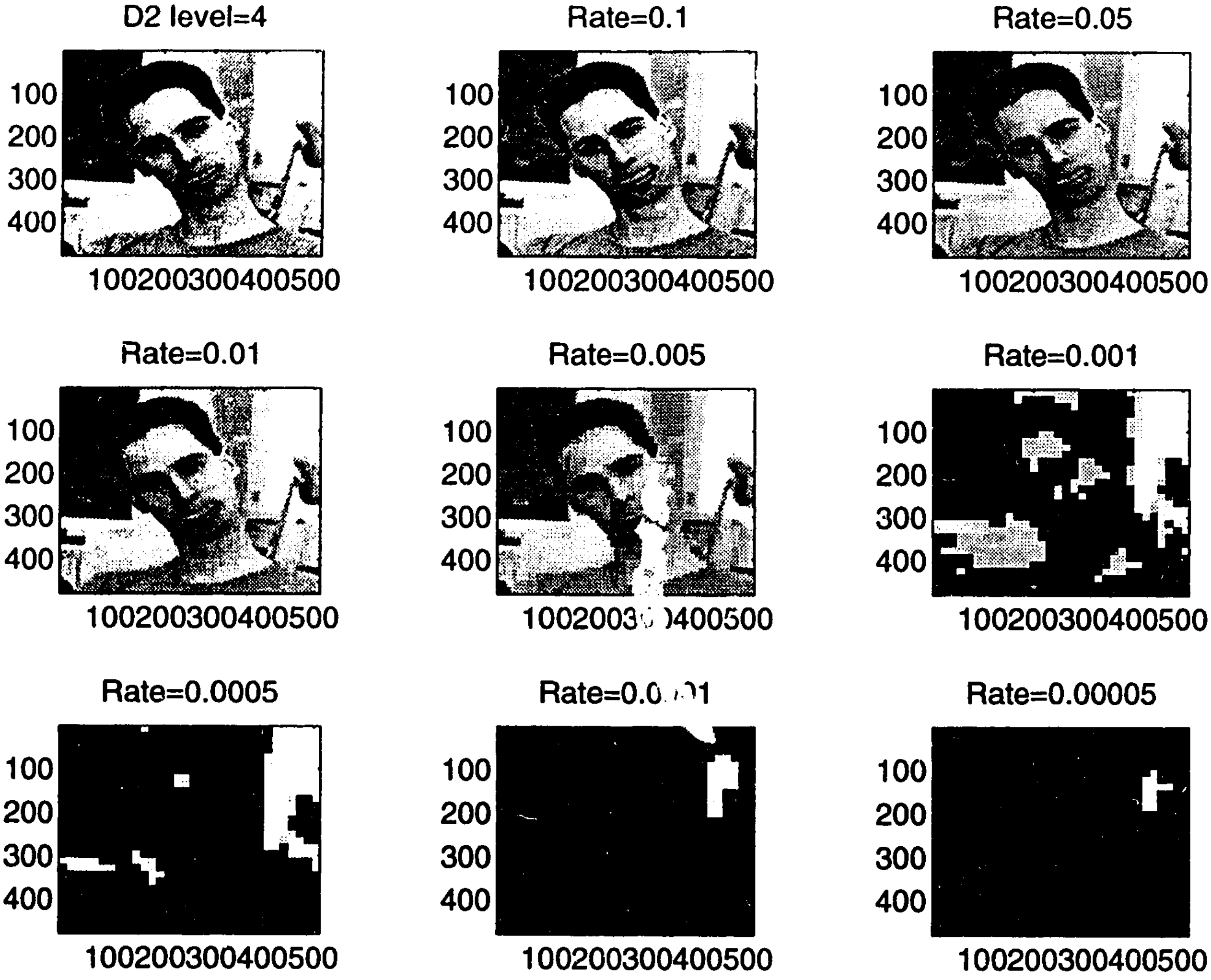
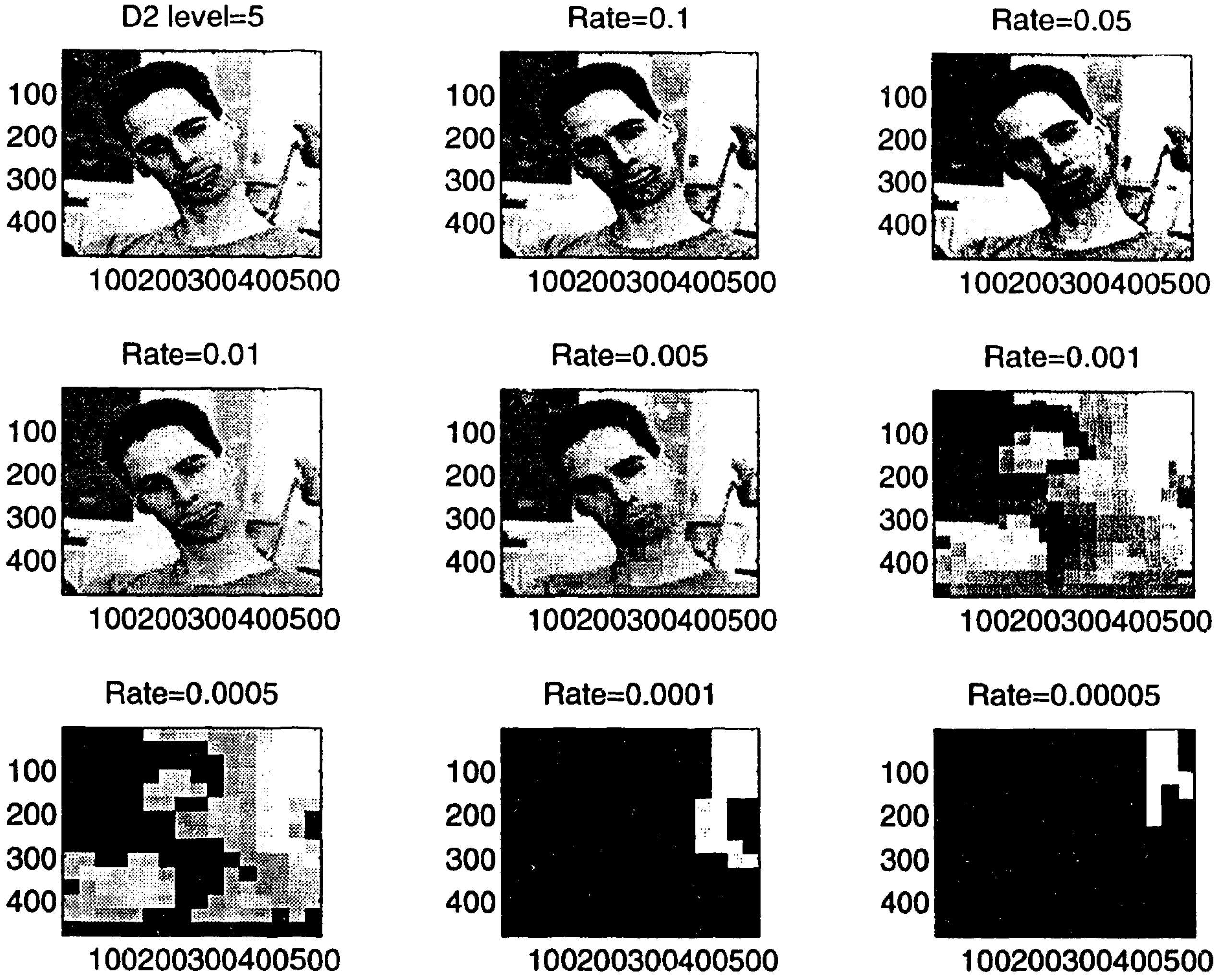


**Figure 6.** (Example 5.2) Scheme I compression with 4 levels of decomposition using the D2 transform at various compression rates. The original image is at top left.


D2 level=5

Rate=0.1

Rate=0.05

Rate=0.01

Rate=0.005

Rate=0.001

Rate=0.0005

Rate=0.0001

Rate=0.00005

**Figure 7.** (Example 5.2) Scheme I compression with 5 levels of decomposition using the D2 transform at various compression rates. The original image is at top left.

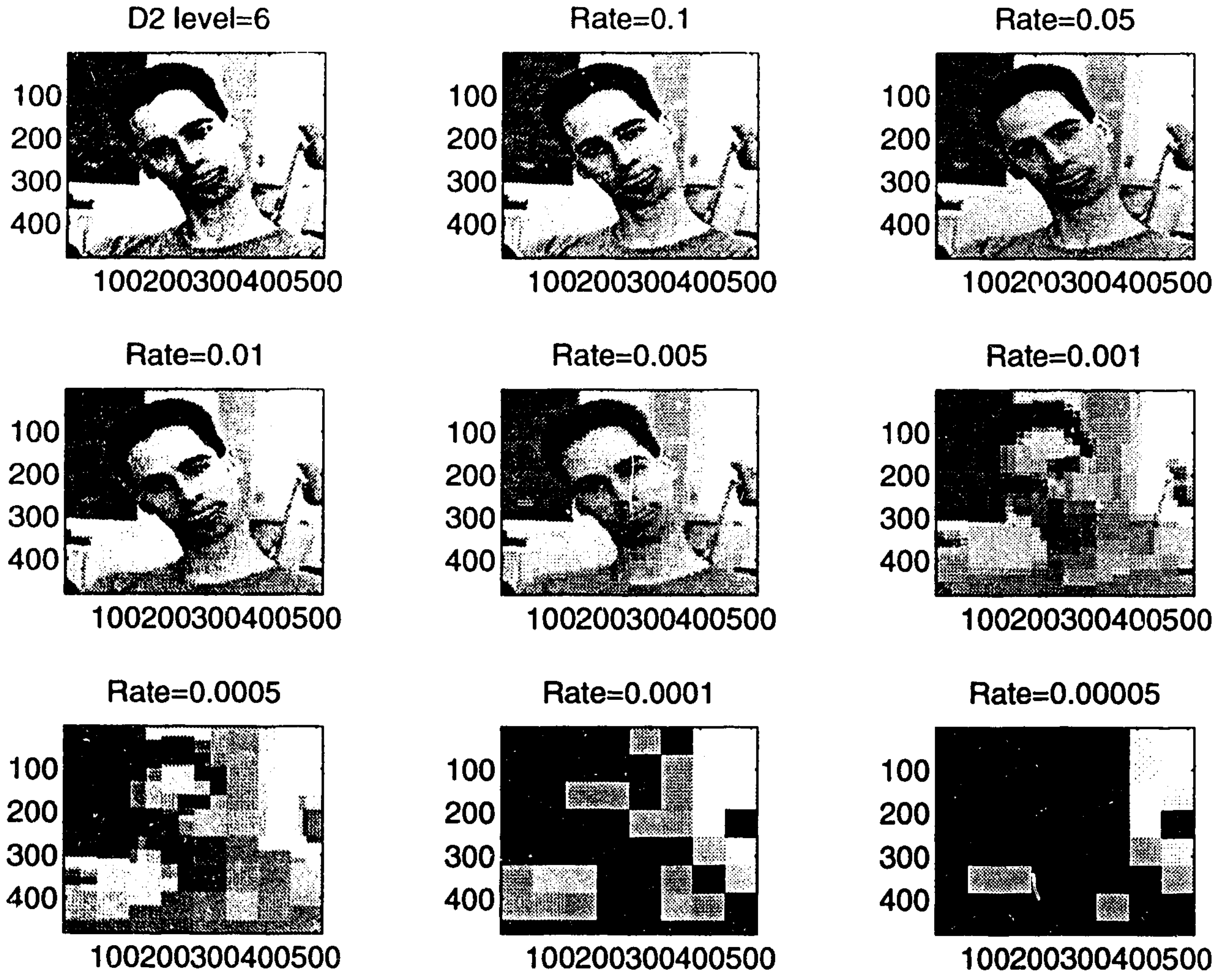
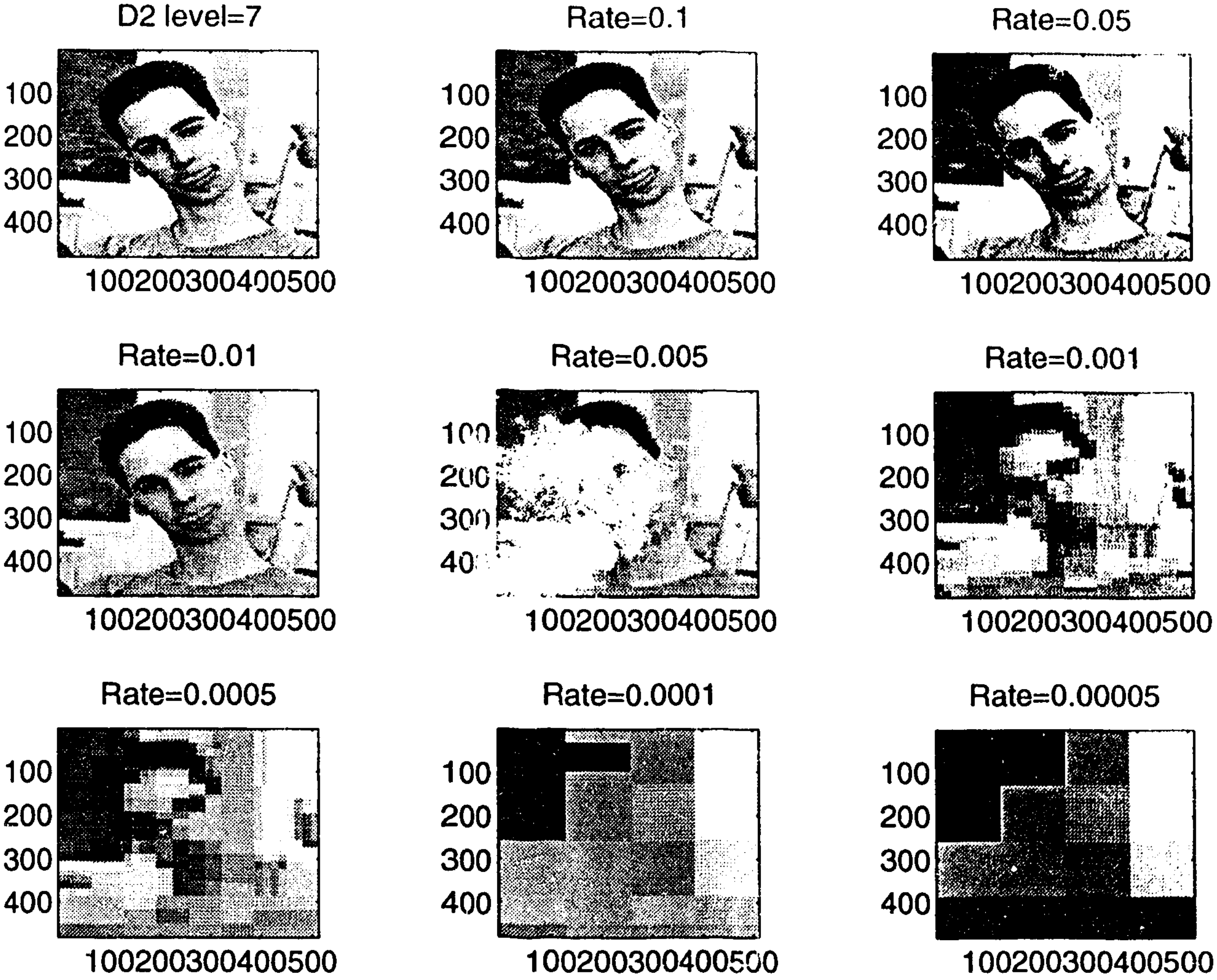


**Figure 8.** (Example 5.2 ) Scheme I compression with 6 levels of decomposition using the D2 transform at various compression rates. The original image is at top left.



D2 level=7

Rate=0.1

Rate=0.05

Rate=0.01

Rate=0.005

Rate=0.001

Rate=0.0005

Rate=0.0001

Rate=0.00005

**Figure 9.** (Example 5.2) Scheme I compression with 7 levels of decomposition using the D2 transform at various compression rates. The original image is at top left.

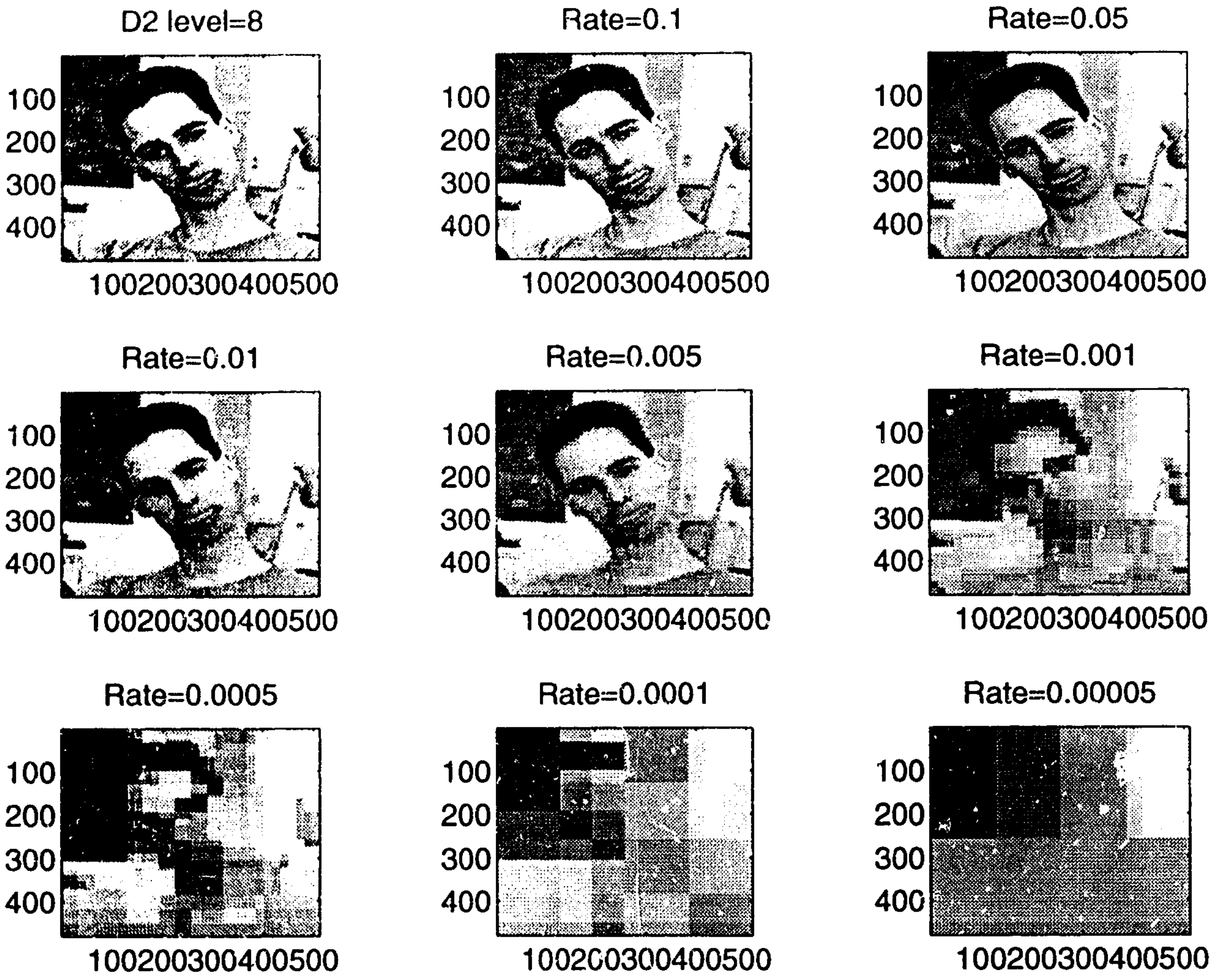


**Figure 10.** (Example 5.2) Scheme I compression with 8 levels of decomposition using the D2 transform at various compression rates. The original image is at top left.

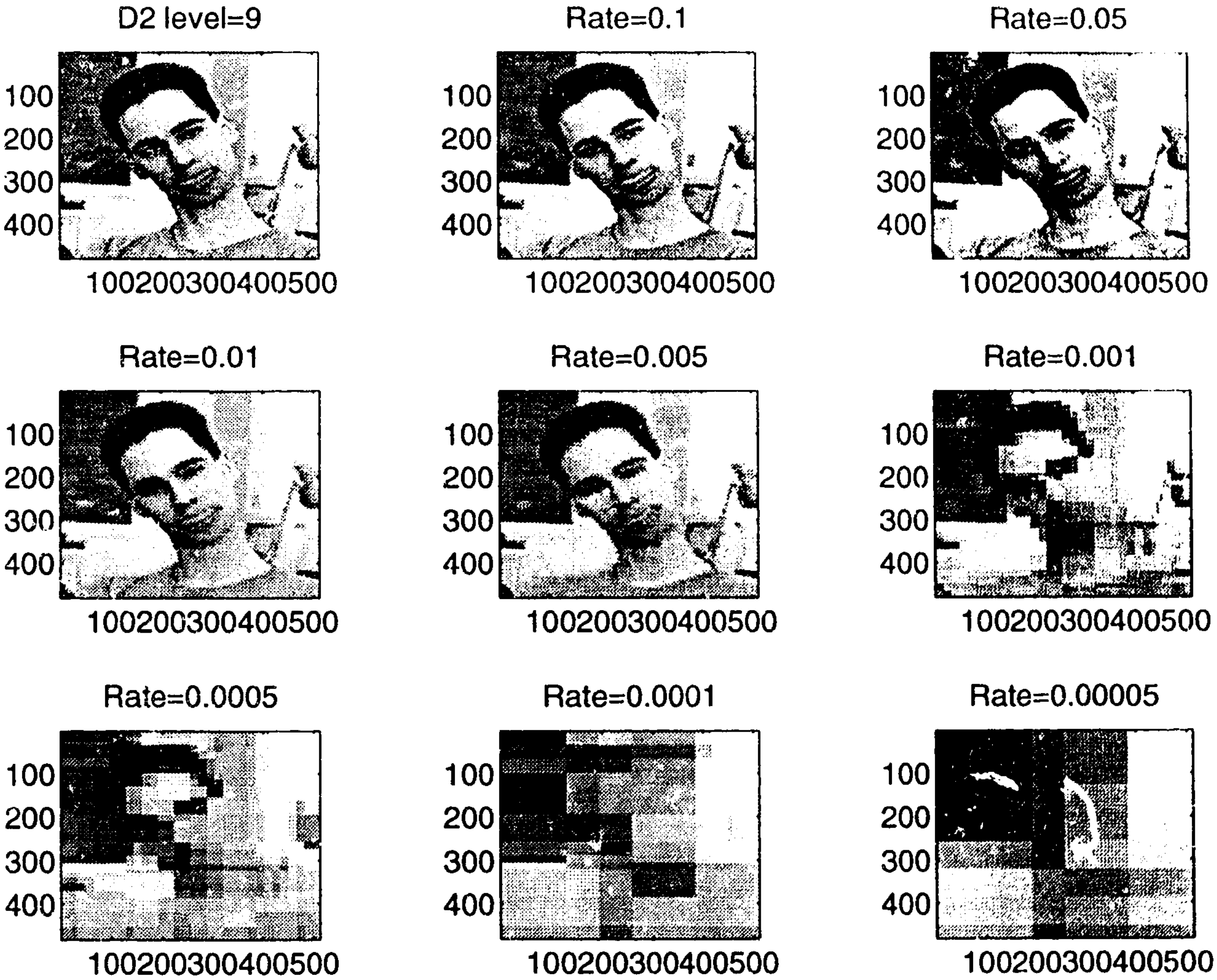


**Figure 11.** (Example 5.2) Scheme I compression with 9 levels of decomposition using the D2 transform at various compression rates. The original image is at top left.

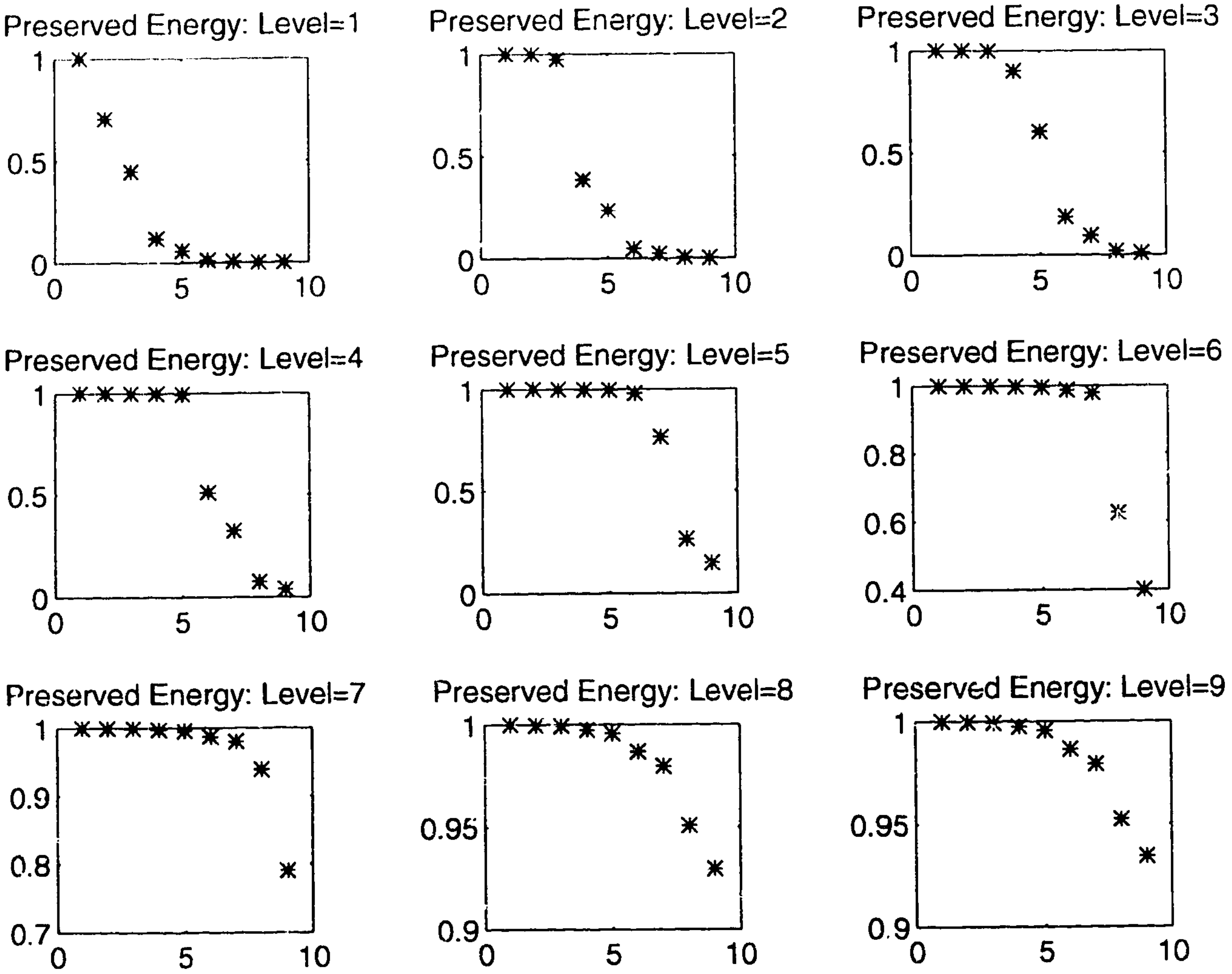


**Figure 12.** (Example 5.2) Percentage of preserved energy of the compressed images: plots here correspond to the levels of decomposition of Figures 3-11. In each plot, points correspond to decreasing rates of compression as shown in the figures.

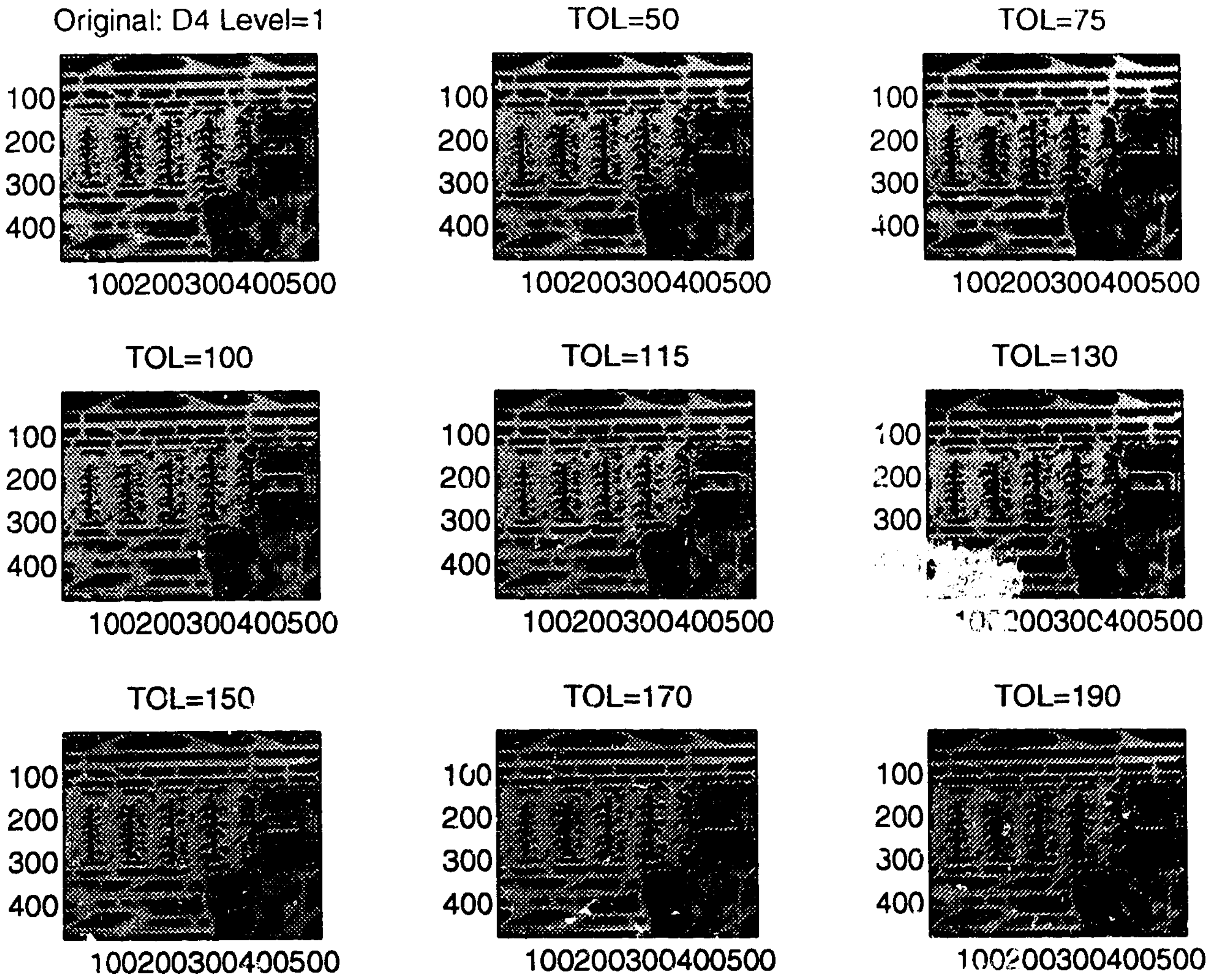


**Figure 13.** (Example 5.3) Scheme II compression with 1 level of decomposition using the D4 transform for increasing tolerance levels. The original image is at top left.

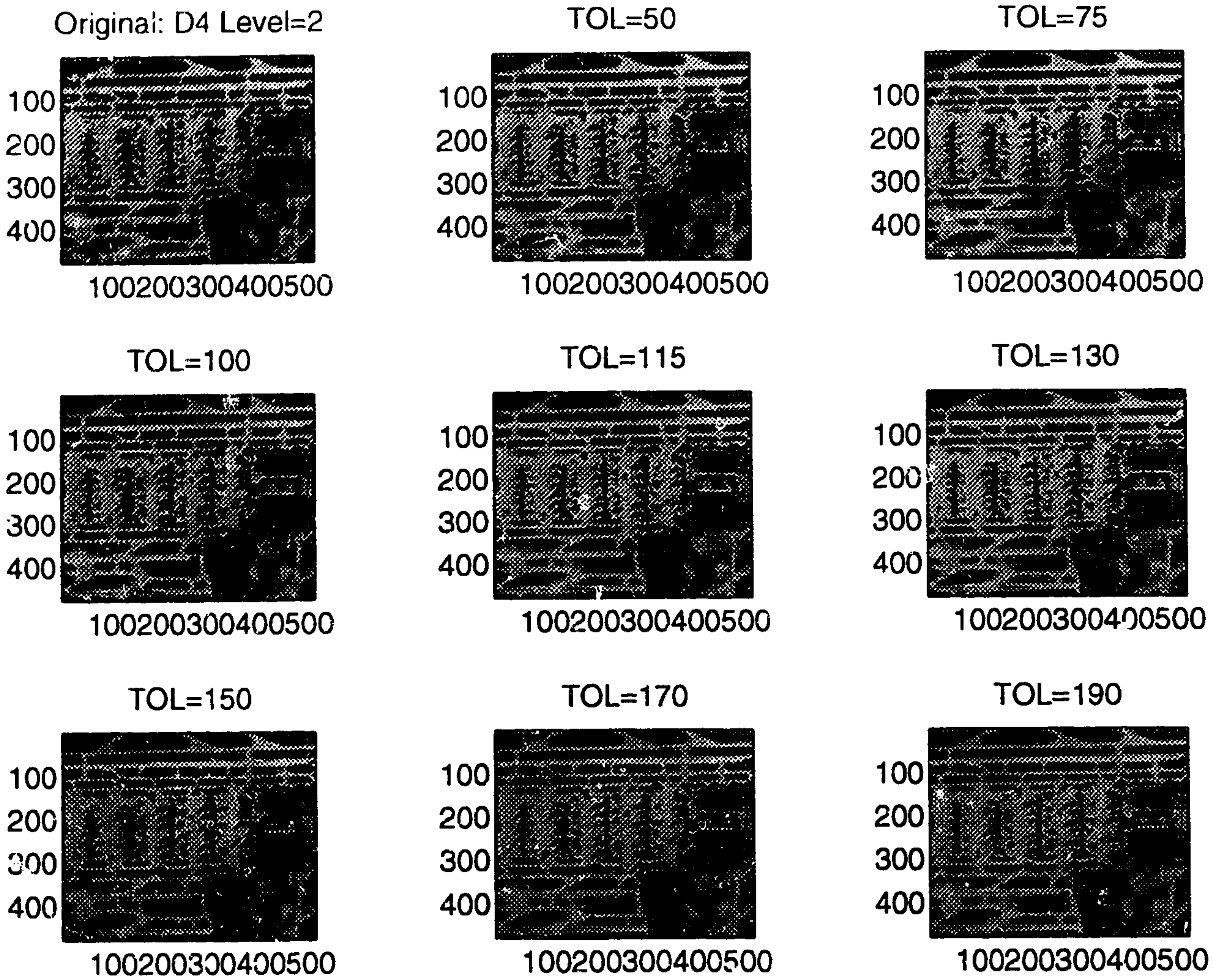


**Figure 14.** (Example 5.3) Scheme II compression with 2 levels of decomposition using the D4 transform for increasing tolerance levels. The original image is at top left.

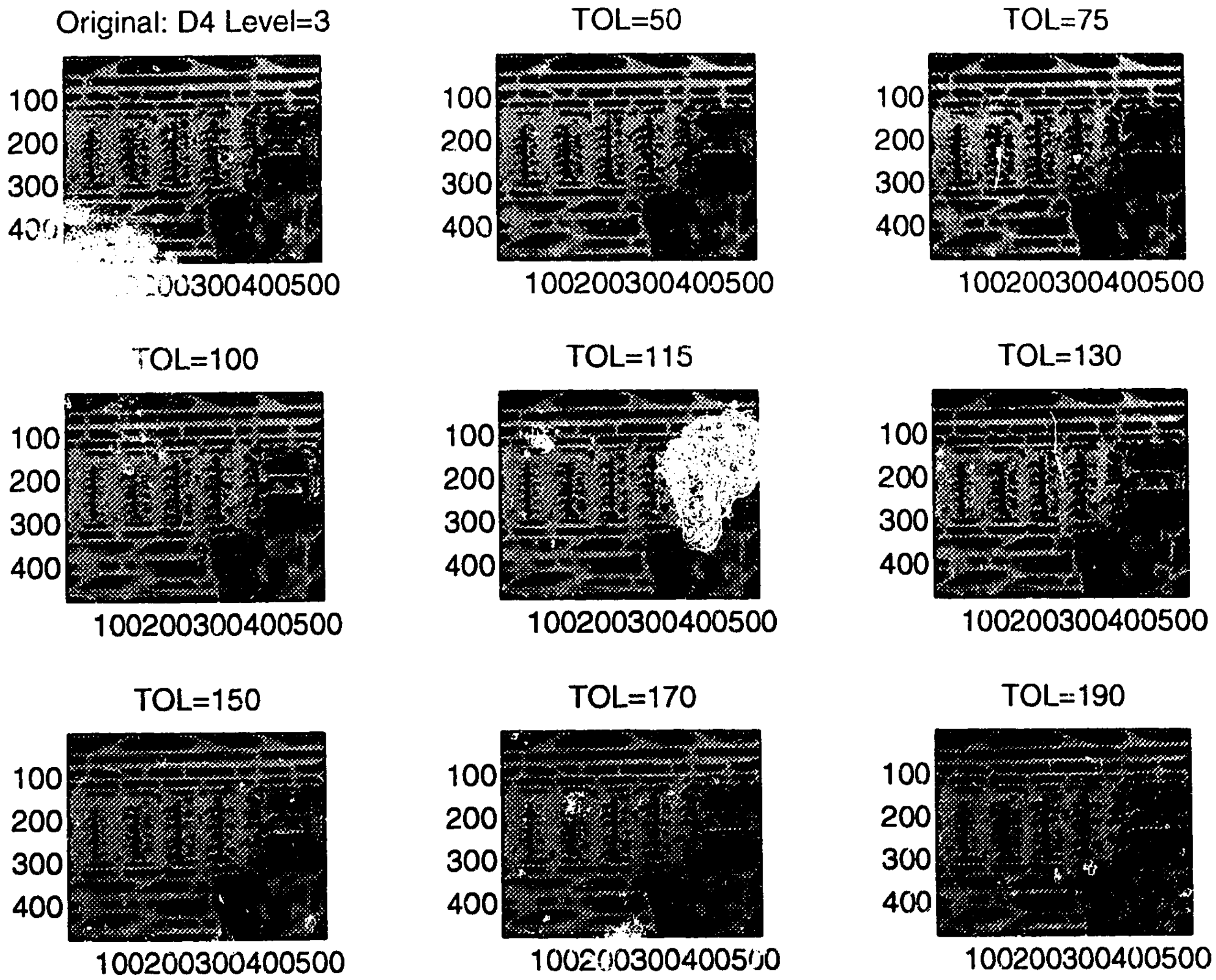


**Figure 15.** (Example 5.3) Scheme II compression with 3 levels of decomposition using the D4 transform for increasing tolerance levels. The original image is at top left.

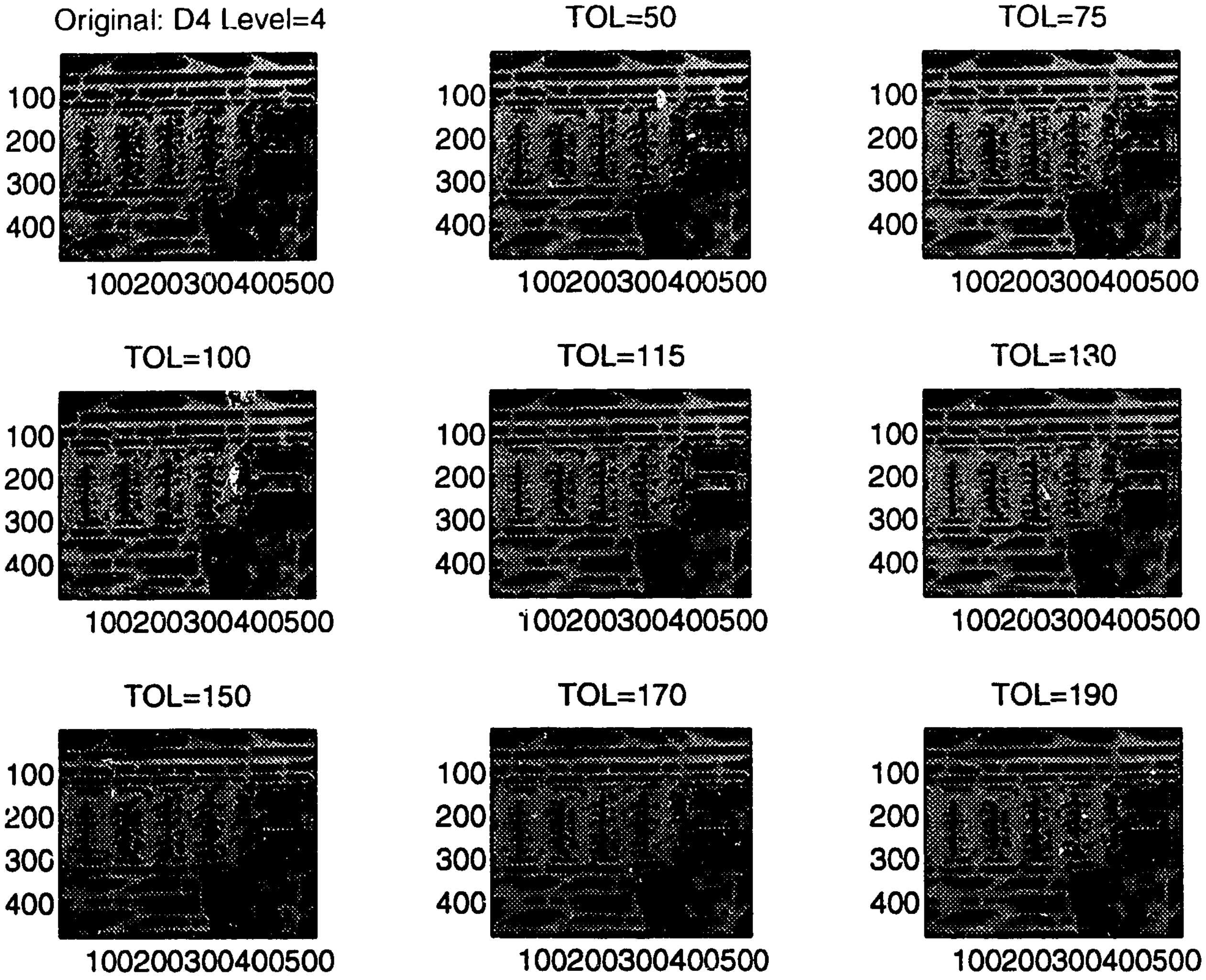


**Figure 16.** (Example 5.3) Scheme II compression with 4 levels of decomposition using the D4 transform for increasing tolerance levels. The original image is at top left.

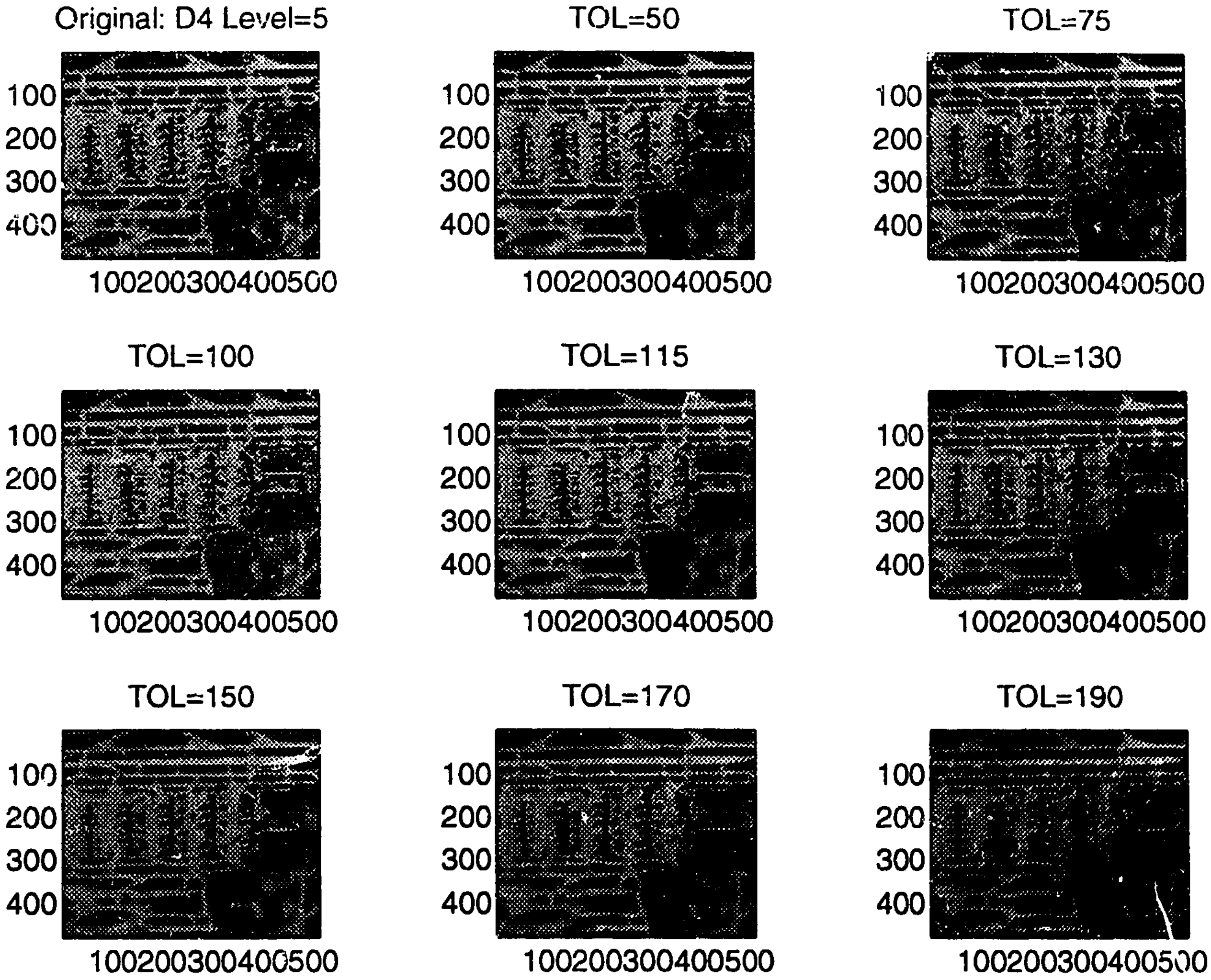


**Figure 17.** (Example 5.3) Scheme II compression with 5 levels of decomposition using the D4 transform for increasing tolerance levels. The original image is at top left.

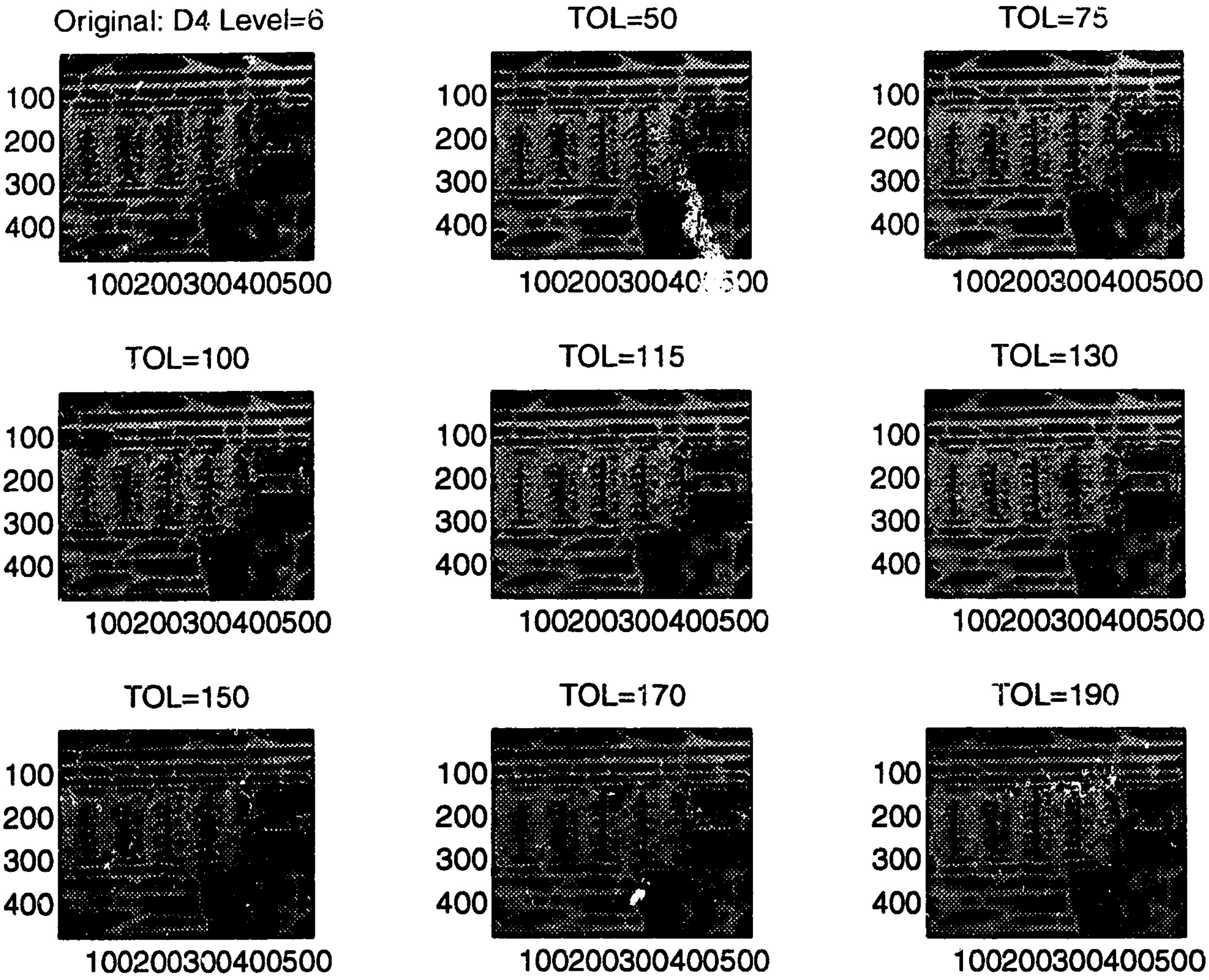


**Figure 18.** (Example 5.3) Scheme II compression with 6 levels of decomposition using the D4 transform for increasing tolerance levels. The original image is at top left.

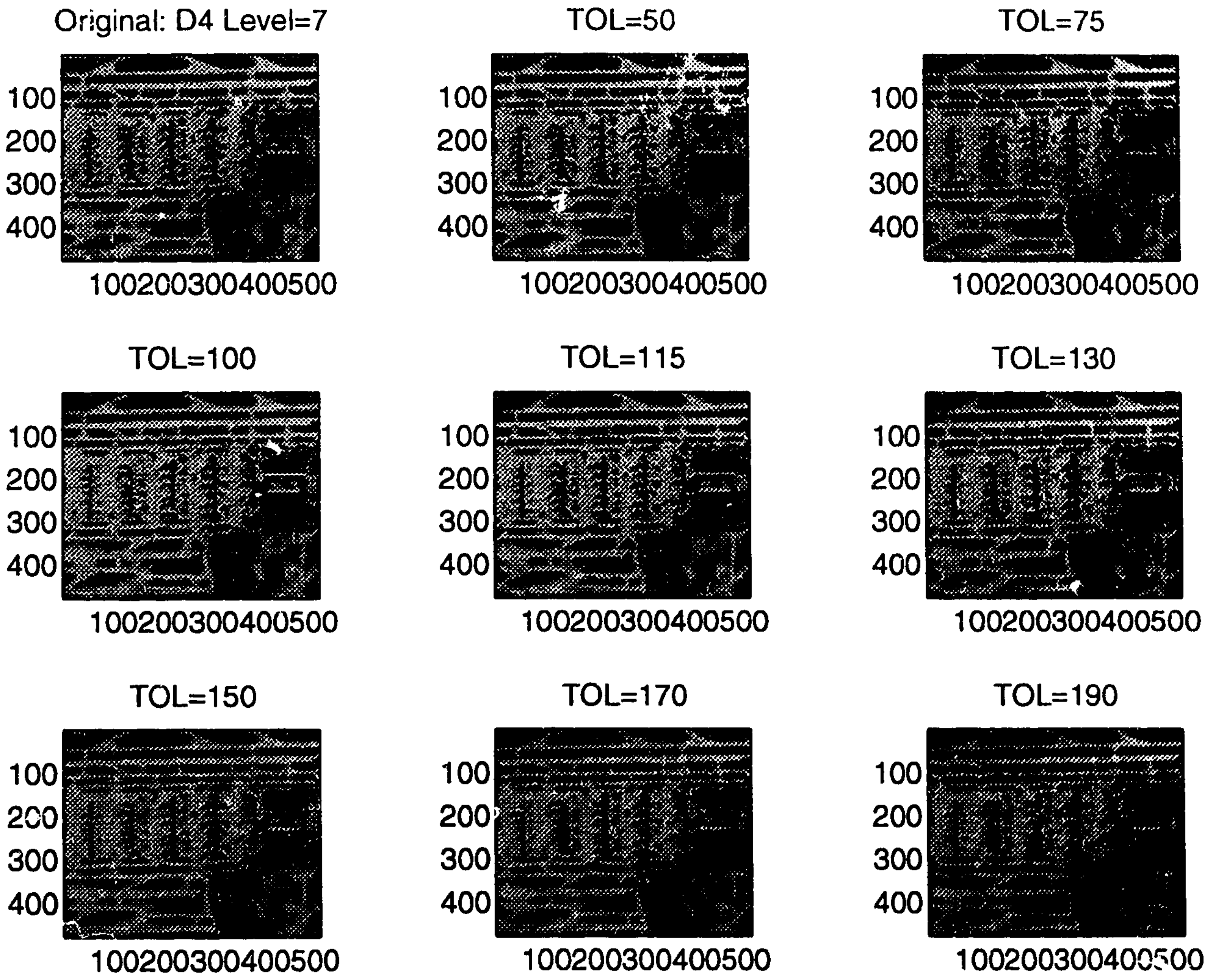


**Figure 19.** (Example 5.3) Scheme II compression with 7 levels of decomposition using the D4 transform for increasing tolerance levels. The original image is at top left.

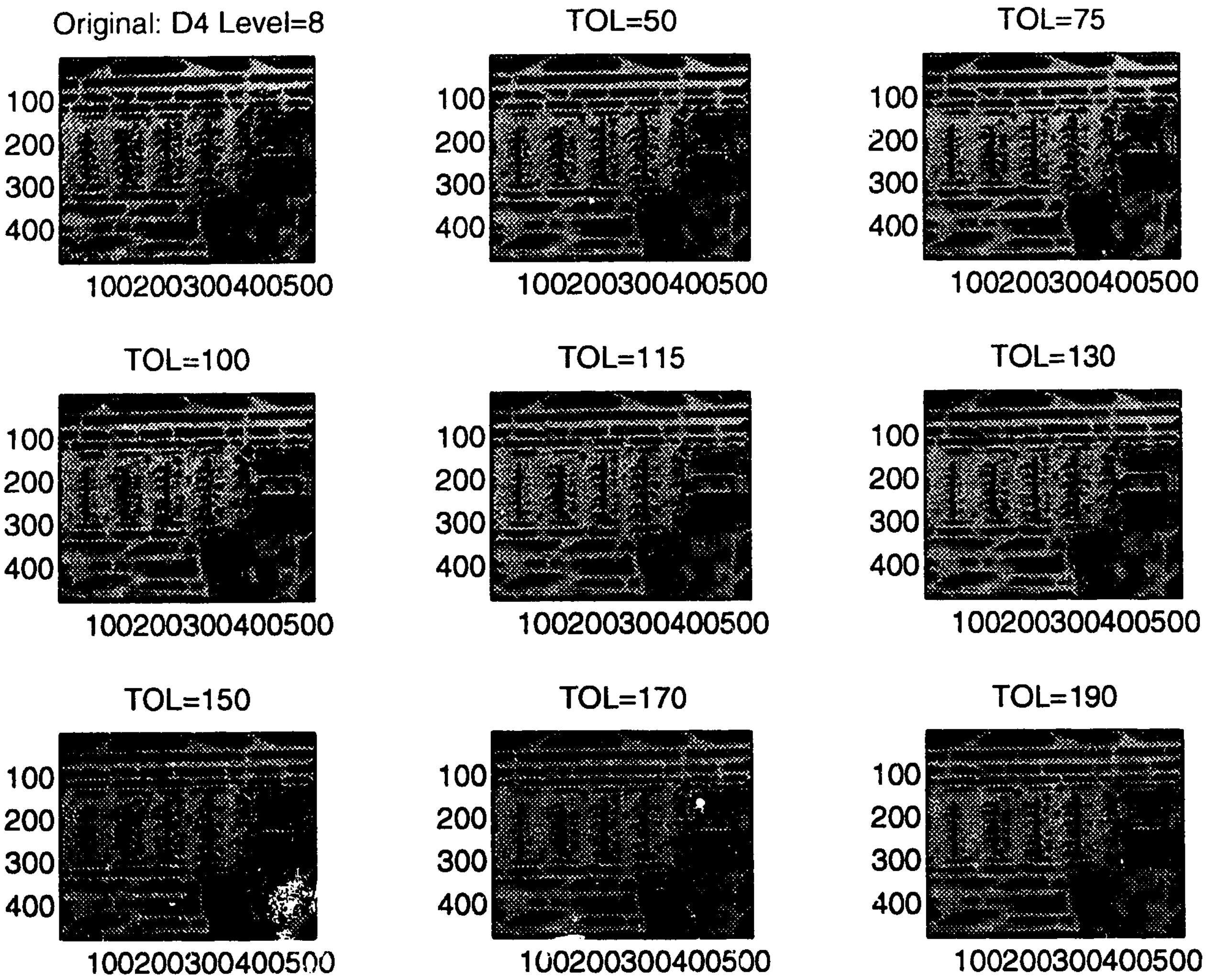


**Figure 20.** (Example 5.3) Scheme II compression with 8 levels of decomposition using the D4 transform for increasing tolerance levels. The original image is at top left.

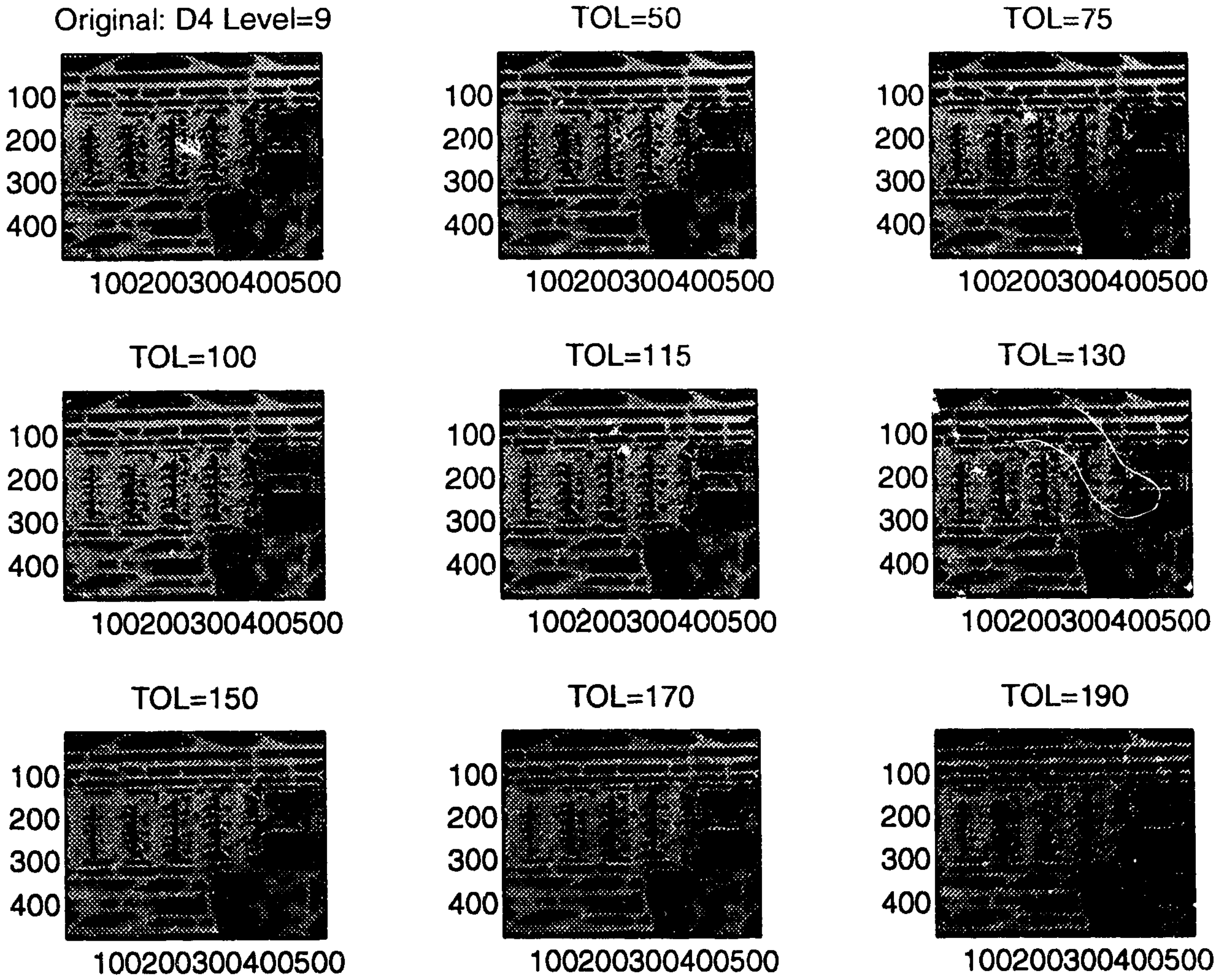


**Figure 21.** (Example 5.3) Scheme II compression with 9 levels of decomposition using the D4 transform for increasing tolerance levels. The original image is at top left.

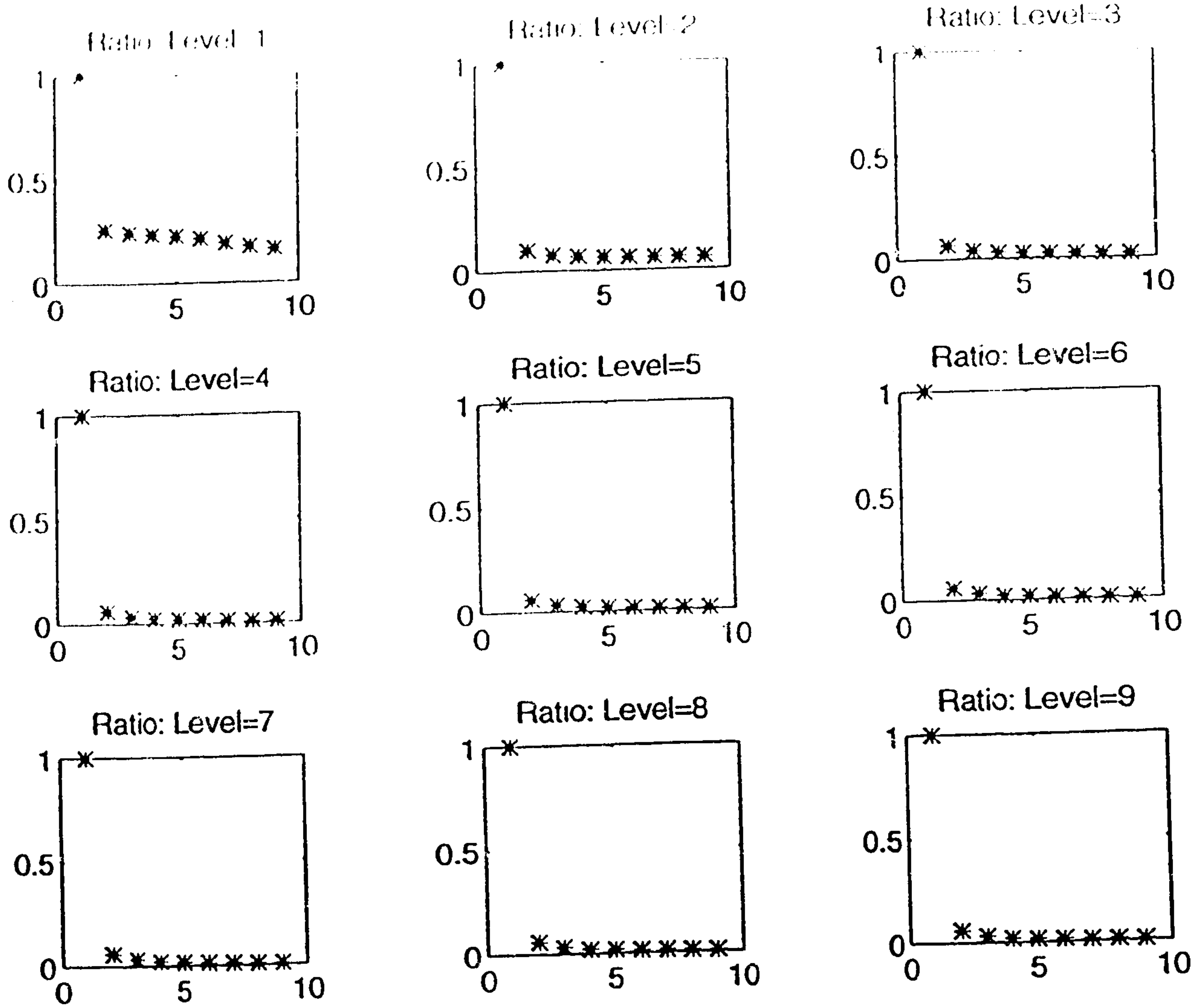


**Figure 22.** (Example 5.3) Compression ratio of the compressed images: plots here correspond to the levels of decomposition as shown in Figures 13-21. In each plot, points correspond to increasing tolerance levels as shown in the figures.

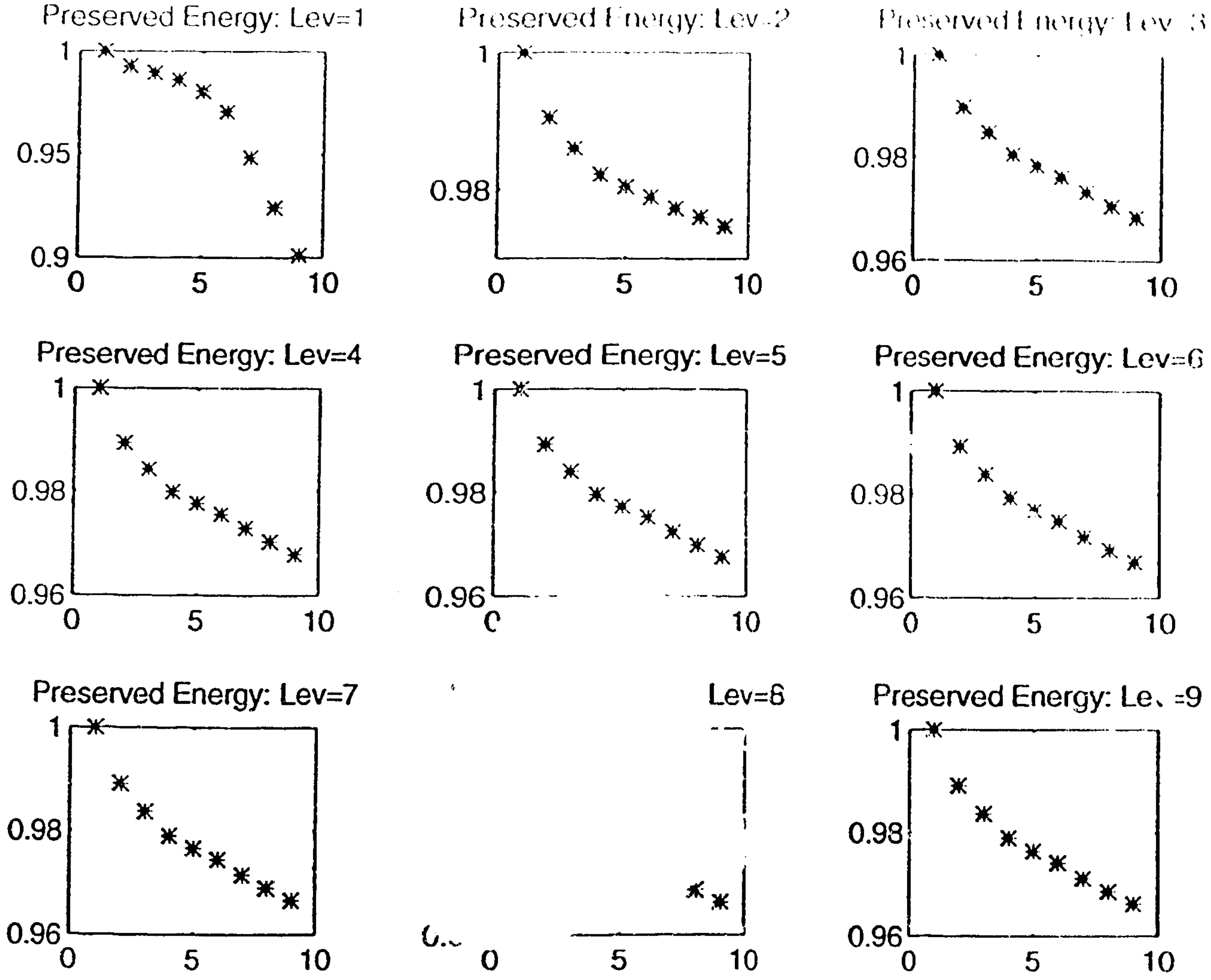


**Figure 23.** (Example 5.3) Percentage of preserved energy of the compressed images: plots here correspond to the levels of decomposition as shown in Figures 13-21. In each plot, points correspond to increasing tolerance levels as shown in the figures.

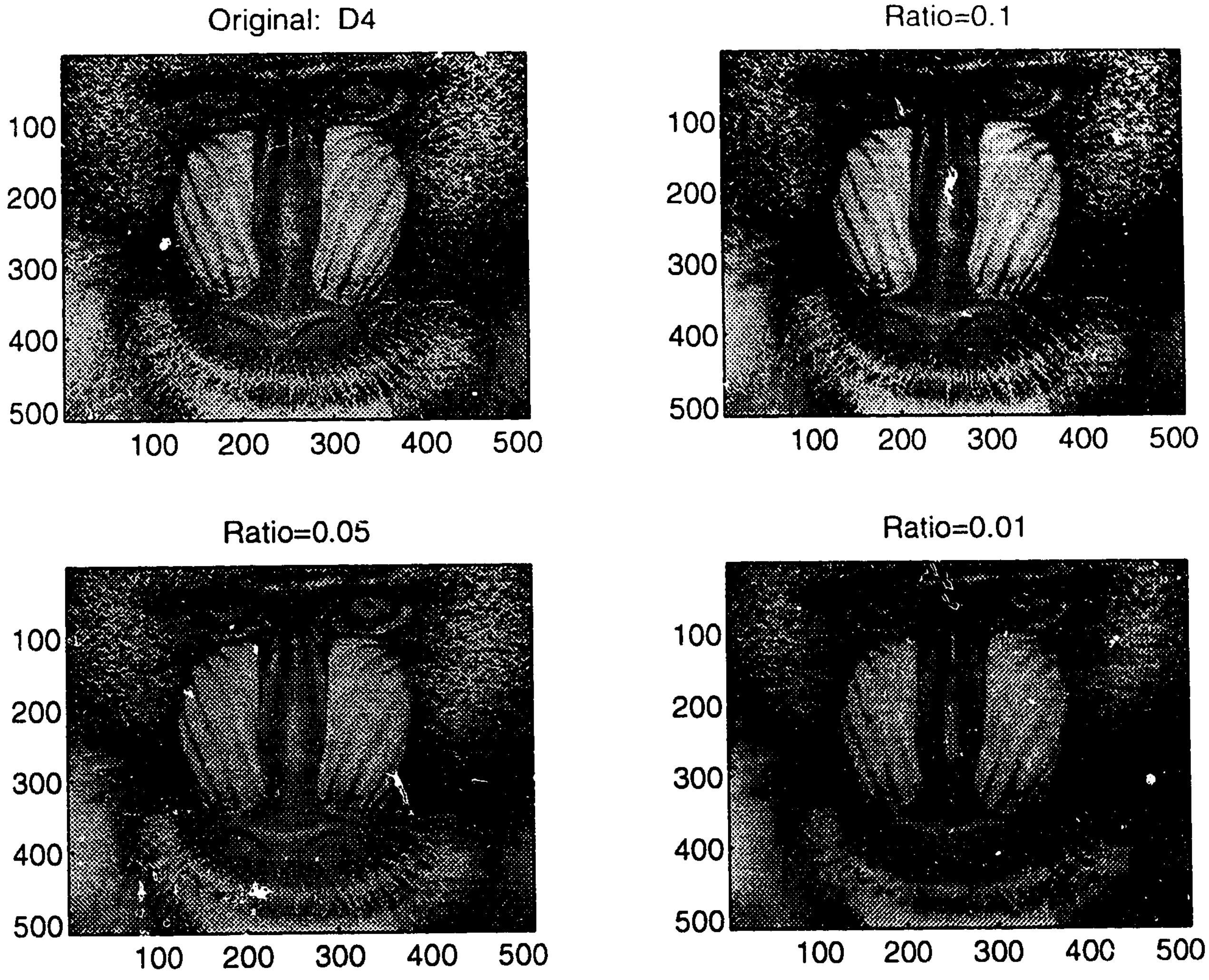


**Figure 24.** (Example 5.4) Scheme I compression with 9 levels of decomposition using the D4 transform at various compression rates. The original image is at top left.

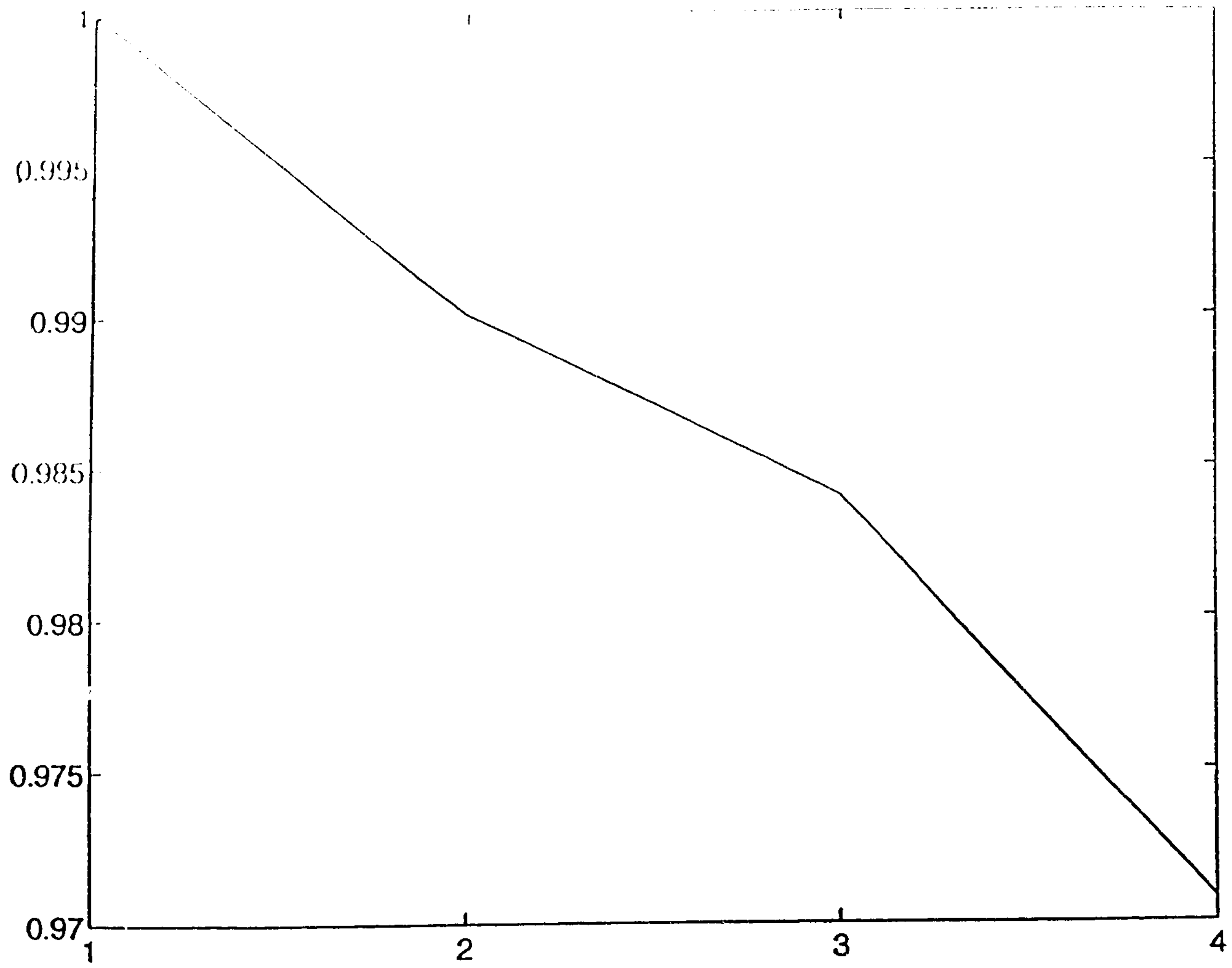


**Figure 25.** (Example 5.4) Percentage of preserved energy of the compressed images: points in this plot correspond to to decreasing rates of compression as shown in Figure 24.

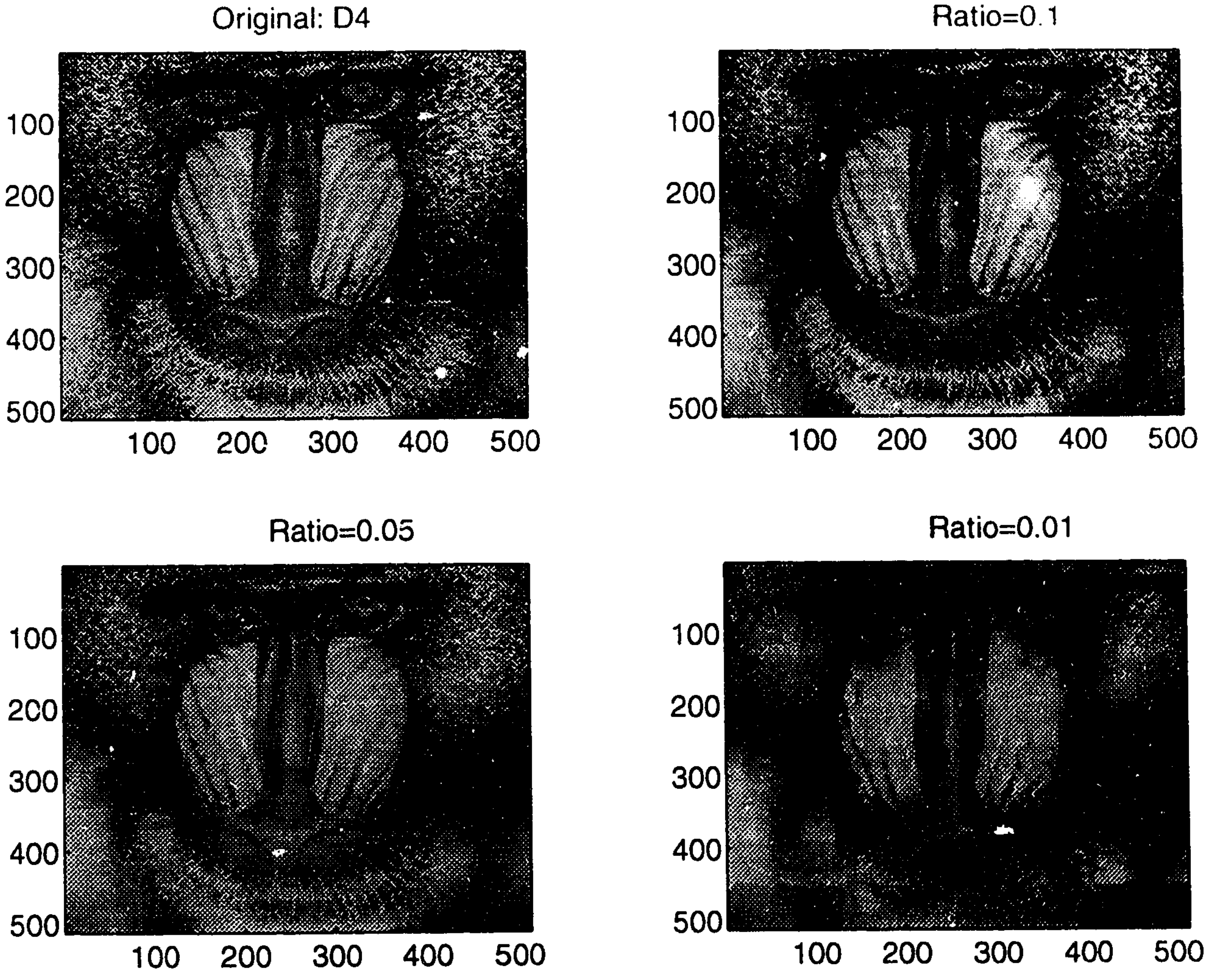


**Figure 26.** (Example 5.4) Pseudo compression with 9 levels of decomposition using the D4 transform at various compression rates. The original image is at top left.

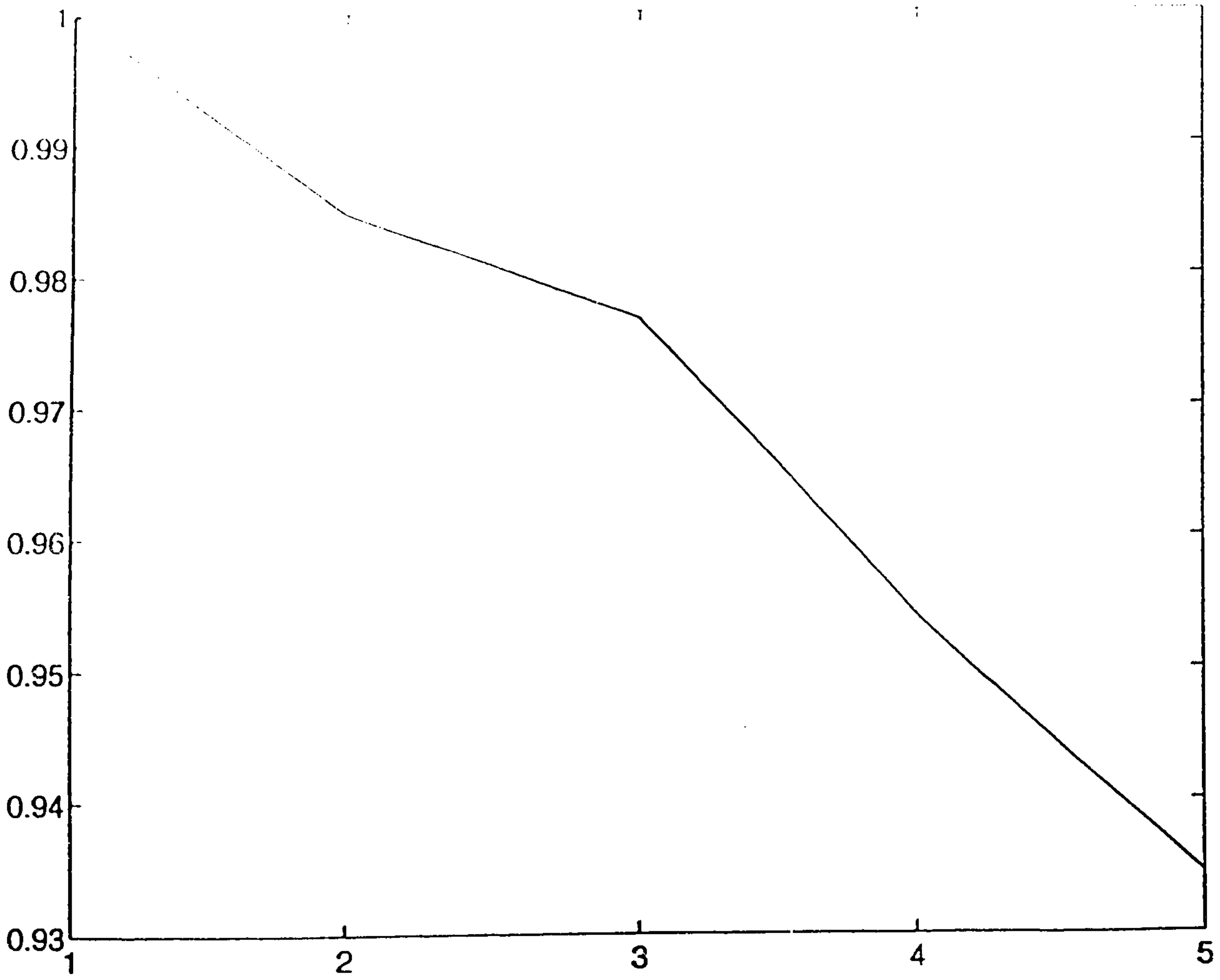


**Figure 27.** (Example 5.4) Percentage of preserved energy of the compressed images: points in this plot correspond to to decreasing rates of compression as shown in Figure 26.

## REFERENCES:

1. G. Beylkin, R. Coifman and V. Rokhlin, Fast wavelet transform and numerical algorithms, Comm. Pure Appl. Math., 44(1991), 141-183.

2. P. J. Burt and E. H. Adelson, The Laplacian pyramid as a compact image code, IEEE Trans. Comm., COM-31(1983), 482-540.

3. R. Coifman and Y. Meyer, Remarques sur l'analyse de Fourier á fenètre, C. R. Acad. Sci. Paris, 312(1991), 259-261.

4. R. Coifman, Y. Meyer, S. Quake and V. Wickerhauser, Signal processing and compression with wavelet packets, M. B. Ruskai etl. eds., Jones & Bartlett, Boston, MA, 1992, 243-264.

5. R. Coifman, Y. Meyer and V. Wickerhauser, Size properties of wavelet packets, Wavelets and their applications, M. B. Ruskai et al. eds., Jones & Bartlett, Boston, MA, 1992, 453-470.

6. I. Daubechies, Orthonormal bases of compactly supported wavelets, Com. Pure Appl. Math., 41(1988), 909-996.

7. I. Daubechies, Orthonormal bases of compactly supported wavelets with finite support - connection with discrete filters, Wavelets: Time-frequency methods in phase space, J. M. Combes et al. eds., Springer-Verlag, Berlin, 1989, 69-76.

8. I. Daubechies, Ten lectures on wavelets, SIAM, Philadelphia, PA, 1992.

9. R. DeVore and P. Lucier, Image compression through wavelet transform coding, IEEE Trans. IT, 2(38), 1992, 719-746.

10. R. DeVore and P. Lucier, Wavelets, Acta Numerica, 1992, 1-56.

11. A. Grossman and J. Morlet, Decomposition of Hardy functions into square integrable wavelets of constant shape, SIAM J. Math., 15(1984), 723-736.

12. C. E. Heil and D. F. Walnut, Continuous and discrete wavelet transforms, SIAM Review, 31(1989), 628-666.

13. S. Mallat, A theory for multiresolution signal decomposition: The wavelet representation, IEEE Trans. Pattern Anal. Machine Intell., 11(1989), 674-693.

14. S. Mallat, Multifrequency channel decomposition of images and wavelet models, IEEE Trans. ASSP, 37(1989), 479-502.

15. S. Mallat and S. Zhong, Complete signal representation with multiscale edges, Robotics Report No. 219, Courant Institute of Mathematical Sciences, New York, NY, 1989.

16. S. Mallat and W. L. Hwang, Singularity detection and processing with wavelets, Robotics Report No. 245, Courant Institute of Mathematical Sciences, New York, NY, 1990.

17. H. S. Malvar, Signal processing with lapped transforms, Artec House, Norwood, MA, 1992.

18. Y. Meyer, Ondelettes, Hermann, New York, 1990.

19. Y. Meyer, Wavelets: Algorithms and applications, SIAM, Philadelphia, PA, 1993.

20. A. Oppenheim and R. W. Schafer, Discrete-time signal processing, Prentice Hall, Englewood Cliffs, New Jersey, 1989.

21. W. Pratt, Digital image processing, John Wiley and Sons, New York, 1992.

22. S. Riemenschneider and Z. Shen, Wavelets and pre-wavelets in low dimensions, JAT, to appear.

23. G. Strang, Wavelets and dilation equations: A brief introduction, SIAM Review, 31(1989), 614-627.

24. A. Zygmund, Trigonometric series, Cambridge University Press, Cambridge, 1959.

## ADDITIONAL REFERENCES

25. B. Jawerth and Wim Sweldens, An overview of wavelet based multiresolution analyses, SIAM Review, 36(1994), 377-412.

The results of this chapter are joint work with S. Riemenschneider.

# Chapter 4

## Numerical Determination of an Unknown coefficient in Semi-linear Parabolic Equations

### §4.0. Introduction

In this chapter we study a finite difference method for approximating the unknown source parameter $p = p(t)$ and $u = u(x, y, t)$ of the following inverse problem. Find $u = u(x, y, t)$ and $p = p(t)$ which satisfy

$$\begin{aligned} &u_t = \Delta u + p(t)u + f(x,y,t), \quad \text{in } Q_T, \\ &u(x,y,0) = \phi(x,y), \qquad (x,y) \in \Omega, \\ &u(x,y,t) = g(x,y,t), \qquad \text{on } \partial\Omega \times [0,T], \end{aligned}$$

subject to the additional specification

$$u(x^*, y^*, t) = E(t), \qquad (x^*, y^*) \in \Omega, \;\; 0 \le t \le T,$$

where $Q_T = \Omega \times (0,T]$, $T > 0$, $\Omega = (0,1) \times (0,1)$, $f, \phi, g$ and $E \neq 0$ are known functions, and $(x^*, y^*)$ is a fixed prescribed interior point in $\Omega$ whose boundary is denoted by $\partial\Omega$. We call this equation semi-linear due to the unknown product $p(t)u$.

If $u$ represents temperature then the problem can be viewed as a control problem of finding the control $p = p(t)$ such that the internal constraint is satisfied.

The problem above and other similar inverse problems of identifying unknown source parameter have been studied by several authors recently [4, 5, 9, 10, 11, 12]. In [5] Cannon, Lin and Wang proposed a predictor-corrector finite difference scheme based upon the Crank-Nicolson method which resulted in a truncation error

$O(\ (\Delta x)^2+(\Delta t)^2\ )$. Numerical results [5] have shown that the procedure is convergent, but there is no theoretical justification.

Here, we will study the backward Euler finite difference scheme. This numerical procedure results in an error $O(\ (\Delta x)^2+(\Delta t)\ )$ and will be shown to be stable in the maximum norm by using the discrete version (modified) of the maximum principle for parabolic finite difference schemes.

This chapter is organized as follows: In Section 4.1 the finite difference scheme is formulated via a transformation, and some necessary preparations are given via some lemmas. In Section 4.2, the stability of our numerical procedure is discussed, and in Section 4.3 some extensions and comments are given. In Section 4.4 we discuss the approximation of $u$ and $p$ in terms of the approximation obtained for the transformed problem. Finally we discuss some numerical computations for several examples which support our theoretical analysis.

Before closing this section, let us define the following transformation:

$$v(x,y,t)=u(x,y,t)\exp\{-\int_0^t p(s)ds\}, \qquad r(t)=\exp\{-\int_0^t p(s)ds\},$$

and

$$u(x,y,t)=\frac{v(x,y,t)}{r(t)}, \qquad p(t)=\frac{-r'(t)}{r(t)}.$$

The transformation: $(u,p)\rightarrow(v,r)$ allows us to eliminate the $p(t)u$ term from the original equation. We see that $(v,r)$ satisfies

$$\begin{aligned}
&v_t=\Delta v+r(t)f(x,y,t), \ \text{ in } Q_T,\\
&v(x,y,0)=\phi(x,y), && (x,y)\in\Omega,\\
&v(x,y,t)=r(t)g(x,y,t), && \text{on } \partial\Omega\times[0,T],
\end{aligned}$$

subject to

$$r(t) = \frac{v(x^*, y^*, t)}{E(t)}, \qquad (x^*, y^*) \in \Omega, \ \ 0 \le t \le T.$$

It clearly follows that $v$ satisfies the following non-local parabolic equation

$$\begin{aligned}
&v_t = \Delta v + v(x^*, y^*, t) F(x, y, t), \ \text{ in } \ Q_T, \\
&v(x, y, 0) = \phi(x, y), \qquad (x, y) \in \Omega, \\
&v(x, y, t) = v(x^*, y^*, t) G \qquad \text{on } \ \partial\Omega \times [0, T],
\end{aligned}$$

where

$$F(x, y, t) = f(x, y, t)/E(t), \qquad G(x, y, t) = g(x, y, t)/E(t).$$

We see from [2, 3, 5] that the above problem is equivalent to the original problem provided that the data is smooth and compatible. Therefore the finite difference scheme is formulated for this problem, and will be shown to be stable in maximum norm. Once $v$ is known numerically the unknown $(u, p)$ can be calculated through the inverse transformation via numerical differentiation. We consider the numerical procedure to be a reasonable one since by controlling the step size in the numerical differentiation, we can demonstrate the convergence of the approximation to $u$ and $p$.

## §4.1. The Backward Euler Scheme

Let $N$ be a positive integer, and $h = \Delta x = \Delta y = 1/N$, $x_i = ih$, $y_j = jh$, where $i, j = 0, \cdots, N$. Let $M > 0$ be a positive integer, and $\tau = T/M$, $t_n = n\tau$, $n = 0, \cdots, M$. For a smooth function $P(x, y) \in C^2(\bar{\Omega})$, we have the following result.

**Lemma 1.1.** *Assume that $P \in C^2(\bar{\Omega})$ and $(i_0, j_0)$ is such that*

$$(x^*, y^*) \in [x_{i_0}, x_{i_0+1}) \times [y_{j_0}, y_{j_0+1}).$$

*Then we have*

$$\begin{aligned} P(x^*, y^*) &= \frac{h-\delta_x}{h}\frac{h-\delta_y}{h}P(x_{i_0}, y_{j_0}) + \frac{h-\delta_x}{h}\frac{\delta_y}{h}P(x_{i_0}, y_{j_0+1}) + \frac{\delta_x}{h}\frac{h-\delta_y}{h}P(x_{i_0+1}, y_{j_0}) \\ &\quad + \frac{\delta_x}{h}\frac{\delta_y}{h}P(x_{i_0+1}, y_{j_0+1}) + O(h^2), \end{aligned}$$

*where*

$$\delta_x = x^* - x_{i_0}, \qquad \delta_y = y^* - y_{j_0}.$$

Proof: It follows from Taylor's expansion that

$$P(x^*, y^*) = \frac{h-\delta_x}{h}P(x_{i_0}, y^*) + \frac{\delta_x}{h}P(x_{i_0+1}, y^*) + O(h^2).$$

Similarly, $P(x_{i_0}, y^*)$ and $P(x_{i_0+1}, y^*)$ can be expanded in terms of $P(x_{i_0}, y_{j_0+1})$, $P(x_{i_0}, y_{j_0})$, and $P(x_{i_0+1}, y_{j_0})$, $P(x_{i_0+1}, y_{j_0+1})$, respectively. □

The obvious truncation of the above equation is used to approximate $v(x^*, y^*, t)$ in our finite difference procedure. The backward Euler finite difference scheme can now be defined. Find $\{v_{i,j}^n\}$ such that

$$\frac{v_{i,j}^n - v_{i,j}^{n-1}}{\tau} = \frac{v_{i+1,j}^n + v_{i-1,j}^n + v_{i,j+1}^n + v_{i,j-1}^n - 4v_{i,j}^n}{h^2} + F_{i,j}^n v_*^{n-1},$$

$$i, j = 1, \cdots, N-1, \qquad n \geq 1,$$

$$v_{i,j}^0 = \phi_{i,j}, \qquad i, j = 0, 1, \cdots, N$$

$$v_{i,j}^n = G_{i,j}^n v_*^{n-1}, \qquad \{i, j\} \bigcap \{0, N\} \neq \emptyset, \quad n > 1,$$

where

$$v_*^n = \frac{h-\delta_x}{h}\frac{h-\delta_y}{h}v_{i_0,j_0}^n + \frac{h-\delta_x}{h}\frac{\delta_y}{h}v_{i_0,j_0+1}^n + \frac{\delta_x}{h}\frac{h-\delta_y}{h}v_{i_0+1,j_0}^n + \frac{\delta_x}{h}\frac{\delta_y}{h}v_{i_0+1,j_0+1}^n, \quad n \geq 1,$$

and where $F_{i,j}^n = F(x_i, y_j, t_n)$ and $G_{i,j}^n = G(x_i, y_j, t_n)$.

It is clear that the above scheme is a semi-implicit finite difference procedure since $v(x^*, y^*, t)$ is approximated using values at the previous level. This scheme results in a truncation error of $O(h^2 + \tau)$, which is the same as the standard backward finite difference scheme for parabolic equations. It is also easy to see that any standard numerical solver for parabolic equation can be used to solve this scheme. For example the alternating direction method could be used very effectively.

**Lemma 1.2.** *The following inequality holds:*

$$|v_*^n| \leq \max_{0 \leq i,j \leq N} |v_{i,j}^n|. \qquad n \geq 0.$$

Proof: It follows directly from our discretized scheme of $v_*^n$ and the definition of $\delta_x$ and $\delta_y$. □

Let us define

$$\Delta_x^+ v_{i,j} = \frac{v_{i+1,j} - v_{i,j}}{h}, \qquad \Delta_x^- v_{i,j} = \frac{v_{i,j} - v_{i-1,j}}{h},$$

and similarly define $\Delta_y^+$ and $\Delta_y^-$. Thus, we have

$$\begin{aligned} \Delta_h^2 v_{i,j} &= (\Delta_x^2 + \Delta_y^2) v_{i,j} = (\Delta_x^- \Delta_x^+ + \Delta_y^- \Delta_y^+) v_{i,j} \\ &= \frac{v_{i+1,j}^n + v_{i-1,j} + v_{i,j+1} + v_{i,j-1} - 4v_{i,j}}{h^2} \end{aligned}$$

and for any $w_{i,j}$ that

$$\begin{aligned} \Delta_h^2 (w_{i,j} v_{i,j}) &= w_{i,j} \Delta_h v_{i,j} + v_{i+1,j} \Delta_x^2 w_{i,j} + v_{i,j+1} \Delta_y^2 w_{i,j} + \Delta_x^+ w_{i-1,j}\ \Delta_x^+ v_{i,j} \\ &\quad + \Delta_x^+ w_{i-1,j}\ \Delta_x^+ v_{i-1,j} + \Delta_y^+ w_{i,j-1}\ \Delta_y^+ v_{i,j} + \Delta_y^+ w_{i,j-1}\ \Delta_y^+ v_{i,j-1}. \end{aligned}$$

It is convenient to state our convergence estimate here.

**Theorem 1.3.** *Assume that* $v \in C^{4,2}(\bar{Q}_T)$. *Then, there exist* $h_0 > 0$ *and* $\tau_0 > 0$, *depending upon the data $f$, $g$, and $E$, $d = dist((x^*, y^*), \partial\Omega)$ and $T > 0$, such that for all $0 < h \leq h_0$ and $0 < \tau \leq \tau_0$, there exists a positive constant $C > 0$, depending on $d$, $T$ and the $C^{4,2}$ norm of $v$, such that*

$$\max_{i,j,n} |v^n_{i,j} - v(x_i, y_j, t_n)| \leq C(h^2 + \tau).$$

Proof: See section 4.2. □

## §4.2. Convergence

This section is devoted to the proof of Theorem 1.3. First let us rewrite the difference scheme as

$$\begin{aligned}
&\frac{v^n_{i,j} - v^{n-1}_{i,j}}{\tau} = \Delta^2_h v^n_{i,j} + F^n_{i,j}\, v^{n-1}_*, && i,j = 1, \cdots, N-1, \quad n \geq 1, \\
&v^0_{i,j} = \phi_{i,j}, \qquad i,j = 0, 1, \cdots, N \\
&v^n_{i,j} = G^n_{i,j}\, v^{n-1}_*, && \{i,j\} \bigcap \{0, N\} \neq \emptyset, \quad n \geq 1,
\end{aligned}$$

The proof of Theorem 1.3 consists of the following several steps.

**STEP 1:** Let $e^n_{i,j} = v^n_{i,j} - v(x_i, y_j, t_n)$. Then we see from the equation of $v$, the above scheme and Taylor expansion that $e^n_{i,j}$ satisfies

$$\begin{aligned}
&\frac{e^n_{i,j} - e^{n-1}_{i,j}}{\tau} = \Delta^2_h e^n_{i,j} + F^n_{i,j} e^{n-1}_* + \epsilon^n_{i,j} && i,j = 1, \cdots, N-1, \quad n \geq 1, \\
&e^0_{i,j} = 0, \qquad i,j = 0, 1, \cdots, N \\
&e^n_{i,j} = G^n_{i,j} e^{n-1}_* + \tau^n_{i,j}, && \{i,j\} \bigcap \{0, N\} = \emptyset, \quad n \geq 1,
\end{aligned}$$

where $\epsilon^n_{i,j}$ and $\tau^n_{i,j}$ are the truncation error induced by the discretizations of the differential equation and boundary conditions respectively.

**Lemma 2.1.** *Assume that $v \in C^{4,2}(Q_T)$ and the data are smooth. Then there are positive constants $K$, and $C_0 = C_0(||v||_{C^{4,2}})$ such that*

$$\max_{0\le i,j\le N,\, 0\le n\le M} |e^n_{i,j}| \le C_0(h^2+\tau), \qquad \max_{0\le i,j\le N,\, 0\le n\le M} |\tau^n_{i,j}| \le C_0(h^2+\tau),$$
$$\max_{0\le i,j\le N,\, 0\le n\le M} |G^n_{i,j}| \le K, \qquad \max_{0\le i,j\le N,\, 0\le n\le M} |F^n_{i,j}| \le K.$$

Proof: The first two inequalities follow from Taylor expansion, and the last two inequalities follows from the smoothness of the data $F$, $G$ and $E \neq 0$. □

**STEP 2:** Let $w(x,y) = 1 + Q\{(x-x^*)^2 + (y-y^*)^2\}$, where $Q > 0$ is a positive constant to be chosen below. Let

$$e^n_{i,j} = w_{i,j}\, Y'^n_{i,j}, \qquad i,j = 0,1,\cdots,N, \quad 0 \le n \le M.$$

Then it follows from the formula for $v^n_*$ that

$$\begin{aligned} e^n_* \;&=\; \frac{h-\delta_x}{h}\frac{h-\delta_y}{h} w_{i_0,j_0}\, Y'^n_{i_0,j_0} + \frac{h-\delta_x}{h}\frac{\delta_y}{h} w_{i_0,j_0+1}\, Y'^n_{i_0,j_0+1} \\ &\quad + \frac{\delta_x}{h}\frac{h-\delta_y}{h} w_{i_0+1,j_0}\, Y'^n_{i_0+1,j_0} + \frac{\delta_x}{h}\frac{\delta_y}{h} w_{i_0+1,j_0+1}\, Y'^n_{i_0+1,j_0+1}, \\ &=\; Y'^n_* + QY'^n_{**}, \end{aligned}$$

where

$$\begin{aligned} Y'^n_{**} = \;&=\; \frac{h-\delta_x}{h}\frac{h-\delta_y}{h}(\delta_x^2+\delta_y^2)\, Y'^n_{i_0,j_0} + \frac{h-\delta_x}{h}\frac{\delta_y}{h}(\delta_x^2+(h-\delta_y)^2)\, Y'^n_{i_0,j_0+1} \\ &\quad + \frac{\delta_x}{h}\frac{h-\delta_y}{h}((h-\delta_x)^2+\delta_y^2)\, Y'^n_{i_0+1,j_0} + \frac{\delta_x}{h}\frac{\delta_y}{h}((h-\delta_x)^2+\delta_y^2)\, Y'^n_{i_0+1,j_0+1}. \end{aligned}$$

**Lemma 2.2.** *We have*

$$|Y'^n_{**}| \le h^2 \max_{0\le i,j,\le N} |Y'^n_{i,j}|, \qquad 0 \le n \le M.$$

Proof: It follows from the formula for $\tilde{Y}^n_{**}$ and the definitions of $\delta_x$ and $\delta_y$. □

Upon using the transformation about $\tilde{Y}^n_{**}$ with some elementary calculations, we find that $\tilde{Y}^n_{i,j}$ satisfies

$$\begin{aligned}
\frac{\tilde{Y}^n_{i,j} - \tilde{Y}^{n-1}_{i,j}}{\tau} &= \Delta_h^2 \tilde{Y}^n_{i,j} + \frac{F^n_{i,j}}{w_{i,j}}(\tilde{Y}^{n-1}_{*} + Q\tilde{Y}^{n-1}_{**}) + \frac{C^n_{i,j}}{w_{i,j}} \\
&\quad + \tilde{Y}^n_{i+1,j}\frac{\Delta_x^2 w_{i,j}}{w_{i,j}} + \tilde{Y}^n_{i,j+1}\frac{\Delta_y^2 w_{i,j}}{w_{i,j}} + \Delta_x^+ \tilde{Y}^n_{i,j}\ \frac{\Delta_x^+ w_{i-1,j}}{w_{i,j}} \\
&\quad + \Delta_x^+ \tilde{Y}^n_{i-1,j}\ \frac{\Delta_x^+ w_{i-1,j}}{w_{i,j}} + \Delta_y^+ \tilde{Y}^n_{i,j}\frac{\Delta_y^+ w_{i,j-1}}{w_{i,j}} + \Delta_y^+ \tilde{Y}^n_{i,j-1}\frac{\Delta_y^+ w_{i,j-1}}{w_{i,j}}. \\
&\qquad\qquad i,j = 1,\cdots,N-1, \quad 1 \le n \le M, \\
\tilde{Y}^0_{i,j} &= 0, \qquad\qquad i,j = 0,1,\cdots,N, \\
\tilde{Y}^n_{i,j} &= \frac{G^n_{i,j}}{w_{i,j}}(\tilde{Y}^{n-1}_{*} + Q\tilde{Y}^{n-1}_{**}) + \frac{\tau^n_{i,j}}{w_{i,j}}, \quad \{i,j\}\bigcap\{0,N\} \ne \emptyset, \quad n \ge 1.
\end{aligned}$$

**Step 3:** Let $\lambda > 0$ and

$$\tilde{Y}^n_{i,j} = e^{\lambda t_n} Z^n_{i,j}, \quad i,j = 0,1,\cdots,N, \quad 1 \le n \le M.$$

Then $Z^n_{i,j}$ satisfies

$$\begin{aligned}
e^{-\lambda\tau}\ \frac{Z^n_{i,j} - Z^{n-1}_{i,j}}{\tau} &= \Delta_h^2 Z^n_{i,j} + e^{-\lambda\tau}\frac{F^n_{i,j}}{w_{i,j}}(Z^{n-1}_{*} + QZ^{n-1}_{**}) + e^{-\lambda t_n}\frac{C^n_{i,j}}{w_{i,j}} - \frac{1 - e^{-\lambda\tau}}{\tau} Z^n_{i,j} \\
&\quad + Z^n_{i+1,j}\frac{\Delta_x^2 w_{i,j}}{w_{i,j}} + Z^n_{i,j+1}\frac{\Delta_y^2 w_{i,j}}{w_{i,j}} + \Delta_x^+ Z^n_{i,j}\ \frac{\Delta_x^+ w_{i-1,j}}{w_{i,j}} \\
&\quad + \Delta_x^+ Z^n_{i-1,j}\ \frac{\Delta_x^+ w_{i-1,j}}{w_{i,j}} + \Delta_y^+ Z^n_{i,j}\frac{\Delta_y^+ w_{i,j-1}}{w_{i,j}} + \Delta_y^+ Z^n_{i,j-1}\frac{\Delta_y^+ w_{i,j-1}}{w_{i,j}}. \\
&\qquad\qquad i,j = 1,\cdots,N-1, \quad 1 \le n \le M, \\
Z^0_{i,j} &= 0. \qquad\qquad i,j = 0,1,\cdots,N, \\
Z^n_{i,j} &= \frac{G^n_{i,j}}{w_{i,j}}(Z^{n-1}_{*} + QZ^{n-1}_{**}) + e^{-\lambda t_n}\frac{\tau^n_{i,j}}{w_{i,j}}, \quad \{i,j\}\bigcap\{0,N\}, \quad n \ge 1.
\end{aligned}$$

**Lemma 2.3.** *For all $i,j, = 0,1,\cdots,N$ for which the differences are defined, we have*

$$|\frac{\Delta_x^2 w_{i,j}}{w_{i,j}}| \le 2Q, \quad |\frac{\Delta_y^2 w_{i,j}}{w_{i,j}}| \le 2Q, \quad |\frac{\Delta_x^+ w_{i-1,j}}{w_{i,j}}| \le 2Q,$$

$$|\frac{\Delta_x^+ w_{i,j}}{w_{i,j}}| \le 2Q, \quad |\frac{\Delta_y^+ w_{i,j-1}}{w_{i,j}}| \le 2Q, \quad |\frac{\Delta_y^+ w_{i,j}}{w_{i,j}}| \le 2Q,$$
$$|\frac{1}{w_{i,j}}| \le \frac{1}{1+Qd^2}, \quad d = dist(\,(x^*,y^*),\partial\Omega),$$

*provided that* $h/2 \le x^*, y^* \le 1 - h/2$.

Proof: It is an elementary argument which we omit. □

**STEP 4:** In this step, we prove that there exist $h_0 = h_0(d,K) > 0$, $\tau_0 = \tau_0(d,K) > 0$, such that for all $0 < h \le h_0$, $0 < \tau \le \tau_0$

$$\max_{0\le i,j \le N,\ 0 \le n \le M} |Z_{i,j}^n| \le C_0(h^2+\tau).$$

Assume that the maximum of $|Z_{i,j}^n|$ is attained at $(i^*,j^*,n^*)$ and that $Z_{i^*,j^*}^{n^*} > 0$. Then there are two cases:

**Case I.** Assume $(i^*,j^*)$ is a boundary point and $M^* = Z_{i^*,j^*}^{n^*}$. Then it follows from the previous step, Lemma 2.1, Lemma 2.2 and Lemma 2.3 that for any $\lambda > 0$,

$$M^* \le \frac{K(1+Qh^2)}{1+Qd^2} M^* + C_0(h^2+\tau),$$

where $C_0$ depends only upon $v$ and $T$. If we select $Q$ and $h$ such that

$$Q = \frac{2\,K}{d^2} \qquad\qquad h \le \frac{d}{\sqrt{2K}}$$

we find $1 - K + Q(d^2-h^2) \ge 1$ and

$$M^* \le \frac{C_0(h^2+\tau)}{1-K+Q(d^2-h^2)} \le C_0(h^2+\tau).$$

**Case II.** If $(i^*,j^*)$ is an interior point, then we take

$$h \le \min\{\frac{1}{2Q}, \quad \frac{d}{\sqrt{2K}}\}.$$

It follows from the discrete maximum principle and STEP 3 that either

$$\frac{1-e^{-\lambda\tau}}{\tau} M^* \le K(M^* + Qh^2M^*) + 4QM^* + C_0(h^2+\tau)$$

or for $\lambda > 0$ large enough, we have for some $\tau_\xi \in (0,\tau)$ and sufficiently small $\tau > 0$ that

$$M^* \leq \frac{C_0(h^2+\tau)}{\lambda e^{-\lambda\tau_\xi} - \{K(1+Qh^2)+4Q\}}.$$

Selecting $\lambda = 2\{K(1+Q)+4Q+1\}$ and $\tau_0 > 0$ such that for $\tau \leq \tau_0$

$$e^{-\lambda\tau_\xi} \geq e^{-\lambda\tau} \geq e^{-\lambda\tau_0} \geq \frac{1}{2} \qquad or \qquad \tau \leq \tau_0 = \frac{\ln 2}{\lambda},$$

we see that

$$\lambda e^{-\lambda\tau_\xi} - \{K(1+Qh^2)+4Q\} \geq 1$$

which in turn implies $M^* \leq C_0(h^2+\tau)$, where $C_0$ depends upon $v$, $T$, $K$ and $d$. By a similar argument we can assume that $Z^{n^*}_{i^*,j^*} < 0$ and obtain the corresponding inequality. □

**STEP 5:** It is easy to see from the above two steps that

$$\max_{0\leq i,j\leq N,\ 0\leq n\leq M} |Y^n_{i,j}| \leq C(h^2+\tau).$$

where $C > 0$ depends upon $v$, $K$, $d$ and $T > 0$. Finally, recalling that $e^n_{i,j} = w_{i,j}Y^n_{i,j}$, we have that

$$\max_{0\leq i,j\leq N,\ 0\leq n\leq M} |e^n_{i,j}| \leq C(h^2+\tau).$$

Therefore, Theorem 1.3 has been proved. □

## §4.3. A Non-uniform Grid Scheme

Let us consider the one-dimensional problem:

$$v_t = \Delta v + v(x^*,t)\,F(x,t), \quad \text{in } Q_T,$$

$$v(x,0) = \phi(x), \qquad x \in \Omega$$

$$v(0,t) = G_1(t)v(x^*,t), \quad v(1,t) = G_2(t)v(x^*,t), \quad x^* \in (0,1), \quad t \in (0,T].$$

Let $h_1 = x^*/k^*$ and $h_2 = (1 - x^*)/(N - k^*)$, where $1 < k^* < N$, and $x_i = ih_1$ for $i = 0, 1, \cdots, k^*$, $x_{k^*} = k^*h$ and $x_i = ih_1$ for $i = k^* + 1, \cdots, N$; and $\Delta t_n = t_n - t_{n-1}$, $n = 1, \cdots, M$. Let $h = \max\{h_1, \ h_2\}$ and $\tau = \max\{ \Delta t_n\}$.

Thus, the finite difference approximation to the solution of this problem can be defined as follows:

$$\begin{aligned}
&\frac{v_i^n - v_i^{n-1}}{\Delta t_n} = \frac{v_{i+1}^n + v_{i-1}^n - 2v_i^n}{h_1^2} + F_i^n \, v_{k^*}^{n-1}, \ \ i = 1, \cdots, k^* - 1, \ \ n \geq 1, \\
&\frac{v_{k^*}^n - v_{k^*}^{n-1}}{\Delta t_n} = 2\frac{h_1 v_{k^*+1}^n + h_2 v_{i-1}^n - (h_1 + h_2)v_i^n}{h_1 h_2 (h_1 + h_2)} + F_{k^*}^n \, v_{k^*}^{n-1}, \\
&\frac{v_i^n - v_i^{n-1}}{\Delta t_n} = \frac{v_{i+1}^n + v_{i-1}^n - 2v_i^n}{h_2^2} + F_i^n \, v_{k^*}^{n-1}, \ \ i = k^* + 1, \cdots, N - 1, \ \ n \geq 1, \\
&v_i^0 = \phi_i, \ i = 0, 1, \cdots, N \\
&v_0^n = G_1^m v_{k^*}^{n-1}, \quad v_N^n = G_2^m v_{k^*}^{n-1}, \quad n \geq 1.
\end{aligned}$$

**Theorem 3.1.** *Assume that $v \in C^{4,2}(\bar{Q_T})$. Then, there exist $h_0 > 0$ and $\tau_0 > 0$ dependent upon $d = min\{x^*, 1 - x^*\}$, and there exists $C > 0$, depending upon $T > 0$ and the $C^{4,2}$ norm of $v$ such that for all $0 < h \leq h_0$ and $0 < \tau \leq \tau_0$ we have*

$$\max_{0 \leq i \leq N, \ 0 \leq n \leq M} |v_i^n - v(x, t_n)| \leq C(h^2 + \tau).$$

Proof: It follows from an argument similar to that given in Section 4.2. The only difference is the auxiliary function $w(x) = 1 + Q(x - x^*)^2$. □

## §4.4. An Error Estimate for $u$ and $p$

We recall our transforms

$$u(x, y, t) = \frac{v(x, y, t)}{r(t)} = \frac{E(t)v(x, y, t)}{v(x^*, y^*, t)}.$$

Also, we recall the results in [5], where $v(x^*, y^*, t) > 0$ for $0 \leq t \leq T$ under various assumptions upon the data. Under those assumptions, there exists a positive number $v_0 > 0$ such that $v(x^*, y^*, t) \geq v_0 > 0$, $0 \leq t \leq T$. Our first application of Lemma 1.1 and Theorem 1.3 are to select $h$ and $\tau$ sufficiently small so that

$$|v(x^*, y^*, t_n) - v_*^n| < \frac{v_0}{2}.$$

As our approximation to $u_{i,j}^n$, we consider

$$W_{i,j}^n = \frac{E^n v_{i,j}^n}{v_*^n}, \qquad i,j = 0,1,\cdots,N, \qquad n = 0,1,\cdots,M.$$

From Theorem 1.3, $v \geq v_0 > 0$, it is an elementary estimation to show that

$$|u_{i,j}^n - W_{i,j}^n| = O(h^2 + \tau), \qquad i,j = 0,1,\cdots,N, \qquad n = 0,1,\cdots,M,$$

for $h$ and $\tau$ sufficiently small. We summarize these results in the following statement.

**Theorem 4.1.** *Let $\phi \geq 0$, $g \geq 0$, $f \geq 0$ and $E > 0$ be such that $\phi \in C^{2+\alpha}(\Omega)$, $g \in C^{1+\alpha/2}(\partial\Omega \times [0,T])$, $E \in C^{1+\alpha/2}([0,T])$, and $f \in C^{\alpha,\alpha/2}(Q_T)$ for some $\alpha$, $0 < \alpha < 1$. Let $\phi(x^*, y^*) = E(0) > 0$ and let the data $\phi, g$ and $f$ satisfy the usual compatibility condition on $\partial\Omega \times \{0\}$. Then, for $h$ and $\tau$ sufficiently small, we have that*

$$|u_{i,j}^n - W_{i,j}^n| = O(h^2 + \tau), \qquad i,j = 0,1,\cdots,N, \qquad n = 0,1,\cdots,M,$$

*holds with $W_{i,j}^n$ defined by*

$$W_{i,j}^n = \frac{E^n v_{i,j}^n}{v_*^n}, \qquad i,j = 0,1,\cdots,N, \qquad n = 0,1,\cdots,M.$$

Proof: According to [5], both of the original and the transformed equations has unique solution which is continuously depend on the data. The conclusion thus follows the analysis preceding the statement of the theorem. □

Turning now to $p = p(t)$, we recall from our transform that

$$\begin{aligned} p(t) &= \frac{-r'(t)}{r(t)} = -\frac{E(t)}{v(x^*,y^*,t)}\frac{\partial}{\partial t}\frac{E(t)}{v(x^*,y^*,t)} \\ &= -\frac{E(t)\frac{\partial}{\partial t}v(x^*,y^*,t) - v(x^*,y^*,t)E'(t)}{v(x^*,y^*,t)E(t)} \\ &= \frac{E'(t)}{E(t)} - \frac{\frac{\partial}{\partial t}v(x^*,y^*,t)}{v(x^*,y^*,t)} \end{aligned}$$

Consequently, an approximation for $p$ involves the numerical computation of $E'(t)$ and $\frac{\partial v}{\partial t}v(x^*,y^*,t)$. As

$$\begin{aligned} \frac{\partial v}{\partial t}(x^*,y^*,t_n) &= \frac{v(x^*,y^*,t_{n+j}) - v(x^*,y^*,t_n)}{j\tau} + O(j\tau) \\ &= \frac{v_*^{n+j} - v_*^n}{j\tau} + O(\frac{h^2}{j\tau} + \frac{\tau}{j\tau} + j\tau) \\ &= \frac{v_*^{n+j} - v_*^n}{j\tau} + O(\frac{h^2}{j\tau} + \frac{1}{j} + j\tau), \end{aligned}$$

we select $j \simeq \sqrt{M}$ and obtain

$$|\frac{\partial v}{\partial t}(x^*,y^*,t_n) - \frac{v_*^{n+j} - v_*^n}{j\tau}| = O(\frac{h^2}{\tau^{1/2}} + \tau^{1/2}).$$

At this point for $\mu > 0$ and fixed, we assume a relationship between $h$ and $\tau$(similar to the stability condition of the forward difference scheme):

$$\frac{h^2}{\tau^{1/2}} = \mu^2\tau^{1/2}.$$

Hence it follows that

$$|\frac{\partial v}{\partial t}(x^*,y^*,t_n) - \frac{v_*^{n+j} - v_*^n}{j\tau}| = O(\tau^{1/2}).$$

Set

$$\rho^n = \frac{E^{n+j} - E^n}{jE^n\tau} - \frac{v_*^{n+j} - v_*^n}{jv_*^n\tau}.$$

Then, observing Theorem 4.1 we can state the following result.

**Corollary 4.2.** *For $h$ and $\tau$ sufficiently small and $h = \mu\tau^{1/2} > 0$ for $\mu$ fixed, we* *have*

$$|p^n - \rho^n| = O(\tau^{1/2})$$

*for* $n = 0, 1, \cdots, M - \sqrt{M}$.

Proof: See the analysis preceding the statement of the corollary. □

## §4.5. Numerical Results

In this section we will present three numerical examples by using the numerical procedure discussed in the previous sections.

**Example 5.1.** Let $u(x,y,t) = \sin(\pi x)\cos(\pi y)e^{-t}$ and $p(t) = t^2 + 1$ be the solutions of the original problem with the initial condition $\phi(x,y) = \sin(\pi x)\cos(\pi y)$, the boundary condition $g(x,0,t) = \sin(\pi x)e^{-t}$, $g(x,1,t) = -\sin(\pi x)e^{-t}$, $g(0,y,t) = g(x,1,t) = 0$, $f(x,y,t) = (-2\pi^2 - t^2 - 2)\sin(\pi x)\cos(\pi y)e^{-t}$ and the additional condition $E(t) = u(x^*, y^*, t) = 1/2e^{-t}$ where $(x^*, y^*) = (0.25, 0.25)$. By a simple calculation, we obtain that $v(x,y,t) = \sin(\pi x)\cos(\pi y)e^{-t^3/2-2t}$, $v(x,y,0) = \phi(x,y)$, $F(x,y,t) = (-4\pi^2 - 2t^2 - 4)\sin(\pi x)\cos(\pi y)$ and $G(x,y,t) = 2\sin(\pi x)\cos(\pi y)$ on $\partial\Omega$. Let $E(v,n) = \max_{i,j} |v^n_{i,j} - v(x_i, y_j t_n)|$, the maximum error on each level for $v$, $E(u,n) = \max_{i,j} |u^n_{i,j} - u(x_i, y_j, t_n)|$, the maximum error on each level for $u$, and $E(p,n) = |p^n - P(t_n)|$. Figures 1-6 show the error we calculated for this example. We find that all error are within the limits predicted by the Theorem 1.3. Also, we noticed that the error for $u$ is usually less than the error for $v$.

**Example 5.2.** Here we take a simple model problem: $\phi(x,y) = x(1-x)$, $u(x,y,t) = 0$ for $x = 0, 1$, $u(x,y,t) = x(1-x)e^{-t}$ for $y = 0, 1$, $f(x,y,t) = 0$ and $E(t) = \frac{3}{16}e^{-t}$ with $(x^*, y^*) = (0.25, 0.25)$. We plot the level surfaces for $v$ and $u$ at $t = 0.5$ and $1.0$, and

$p(t)$ from 0 to 1. These results are listed in Figures 7-9.

**Example 5.3.** Here we take another simple model problem: $\phi(x,t) = 1$, $u(x,y,t) = 1$ for $y = 0, 1, x = 1$, $u(x,y,t) = 1 + y(1-y)\sin(5\pi t)$ for $x = 0$, $f(x,y,t) = 0$ and $E(t) = 2 - \cos(5\pi t)$ with $(x^*, y^*) = (0.25, 0.25)$. As we expected, $p$ is periodic due to the periodicity of the boundary values. We plot the level surfaces for $v$ and $u$ at $t = 0.5$ and $1.0$, and $p(t)$ from 0 to 1. These results are listed in Figures 10-12.

**Remark 5.4.** In our computations of the example 5.1, the restriction on the step sizes of Corollary 5.2 is not satisfied for a particular $\mu$, but $\mu$ varies from 1 to 1.9. This shows a flexibility and robu 'ness of our numerical procedure.

**Ren. .rk 5.5.** In example 5.1, it is easy to see that the error for $p$ in all three cases satisfy $|p^n - p^n| \leq C\sqrt{\tau}$ with $C = 5$. This has verified our theoretical prediction in Corollary 5.2.

**Remark 5.6.** Our numerical procedure is semi-implicit and has no restriction on the step sizes, but numerical differentiation estimate in Corollary 5.2 requires the same step sizes restriction as the standard explicit forward Euler scheme. As numerical differentiation is concerned, it might be better if the forward Euler method is used in the computation of $v$. The reason for using the backward Euler method in this chapter is that it allows us a certain freedom to choose the step size according our needs as stated in Remark 5.4 that $\mu$ can vary in certain range if it is not too small or too large. This will become more important and necessary in higher dimensions.

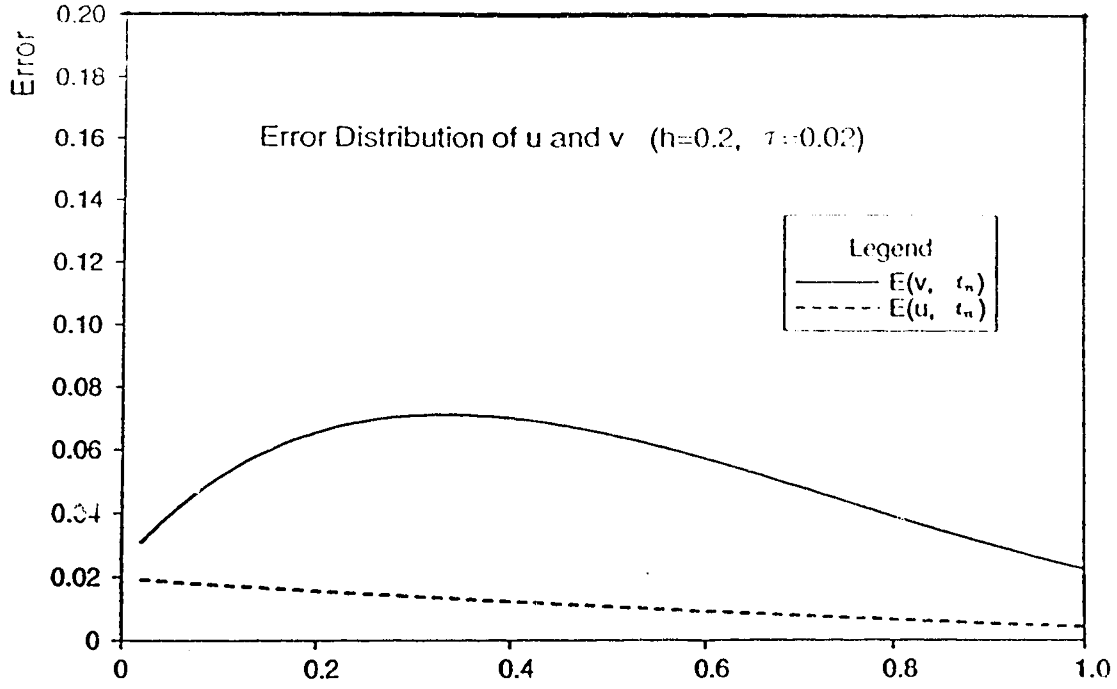


**Figure 1.**

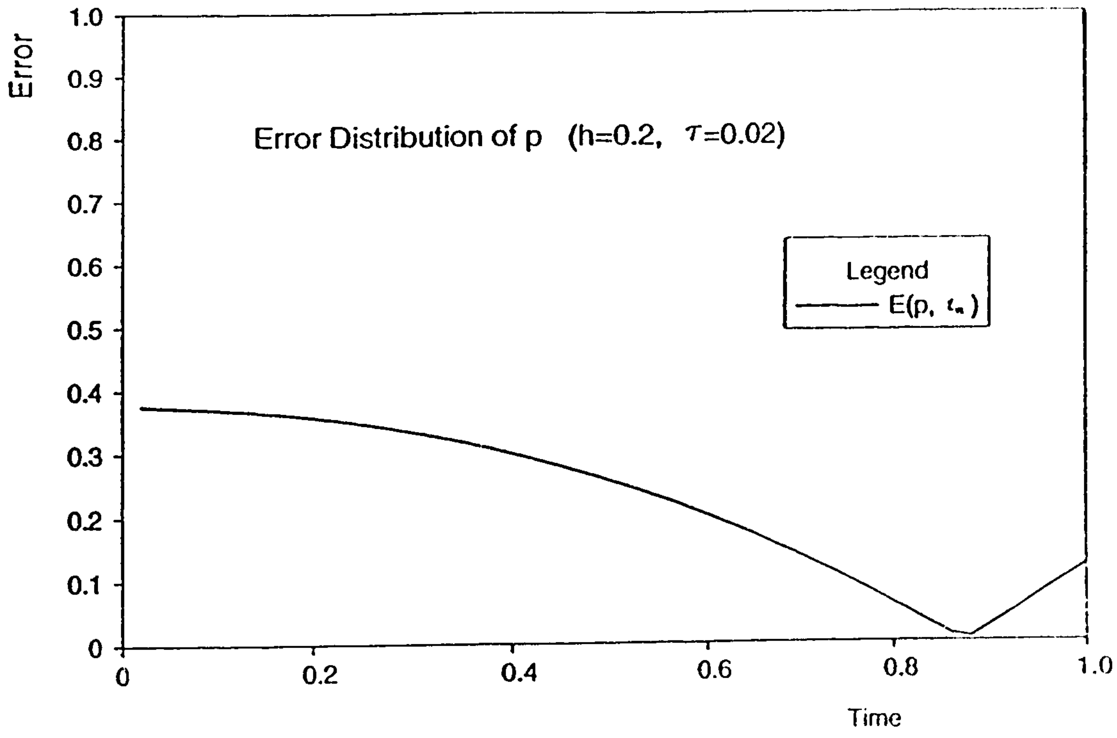


**Figure 2.**

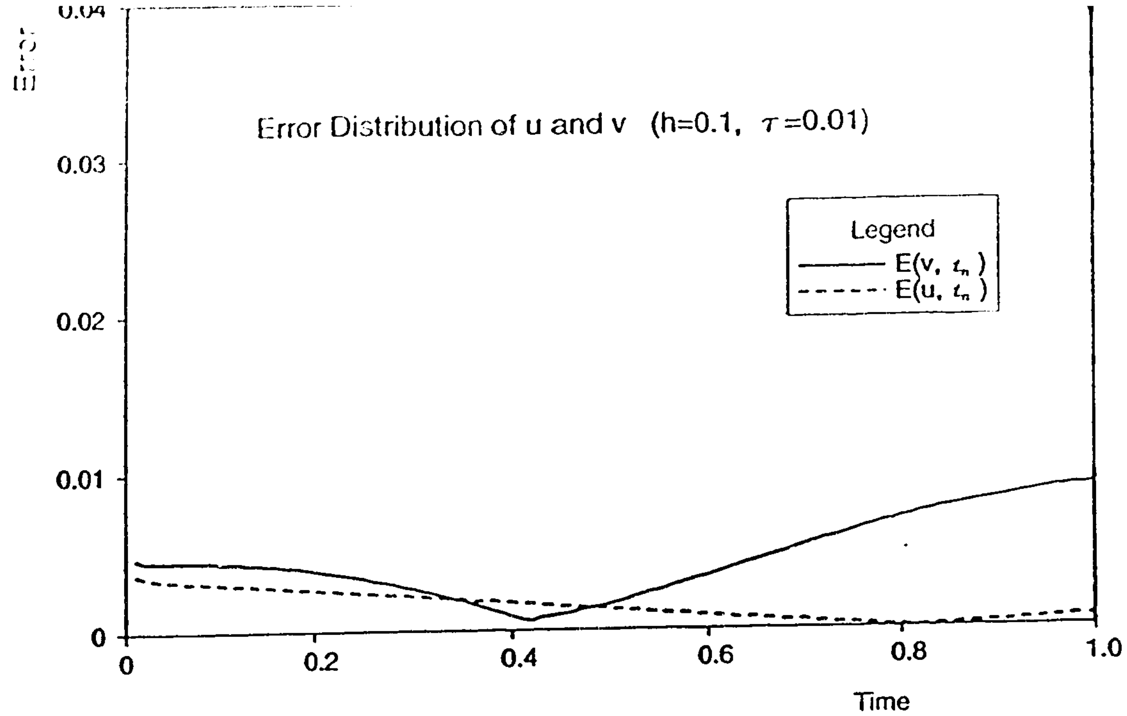


**Figure 3.**

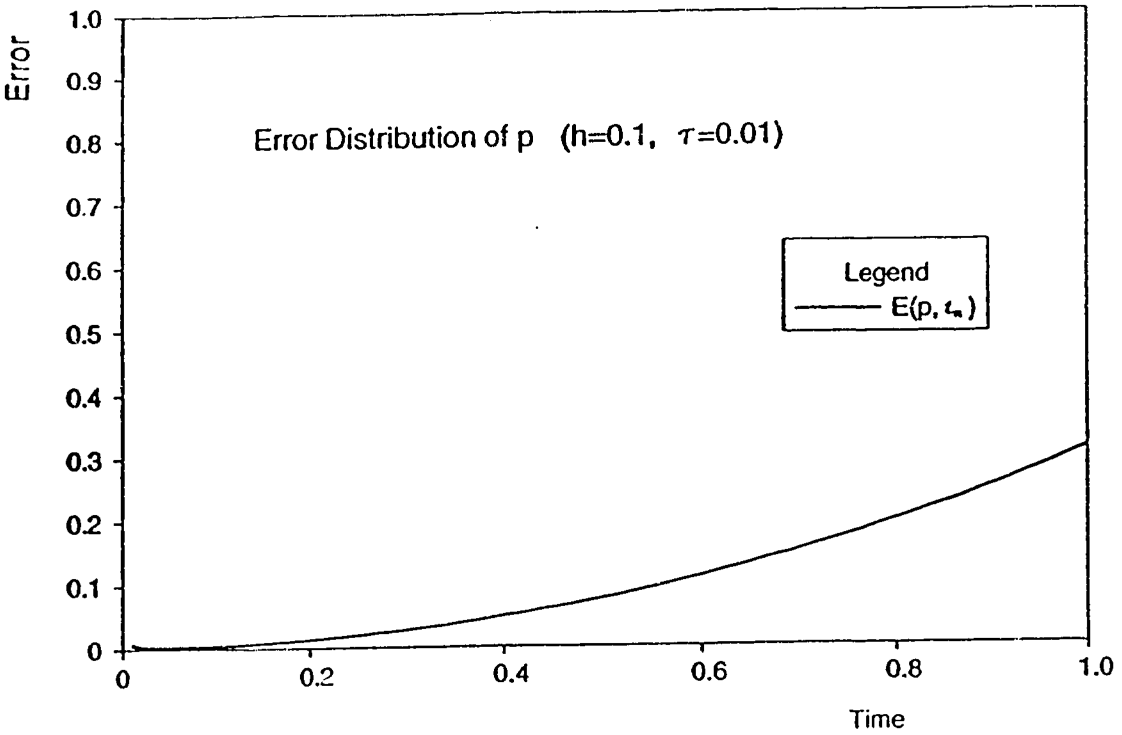


**Figure 4.**

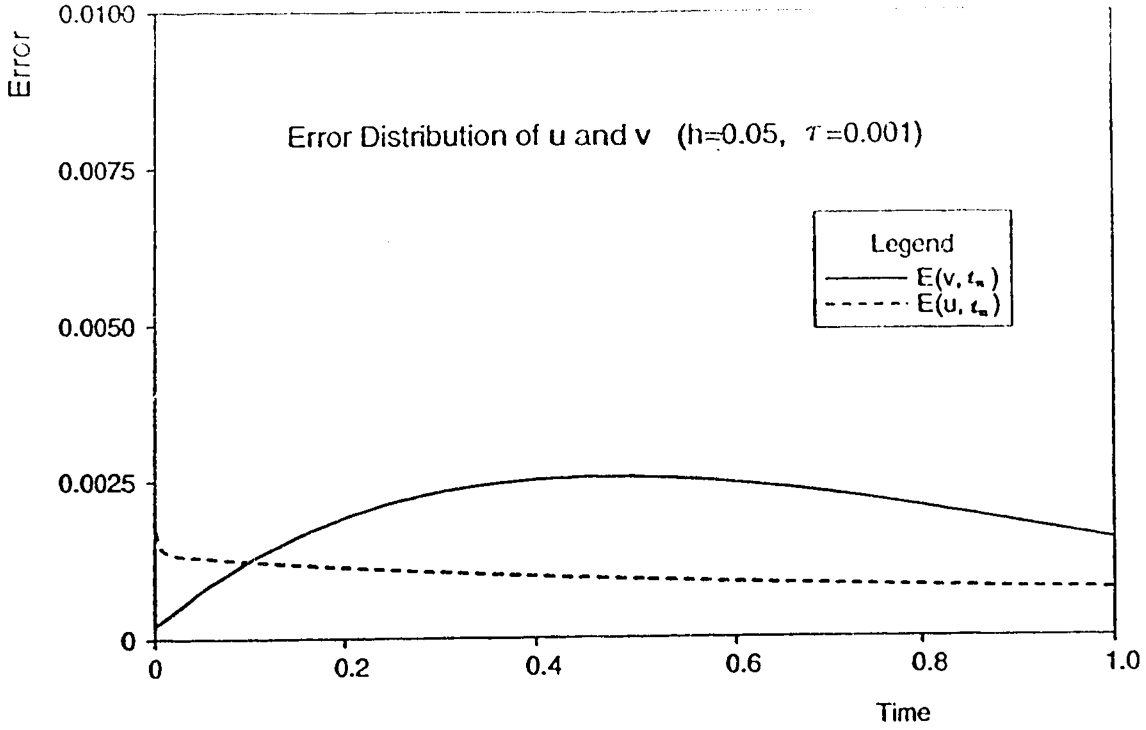


**Figure 5.**

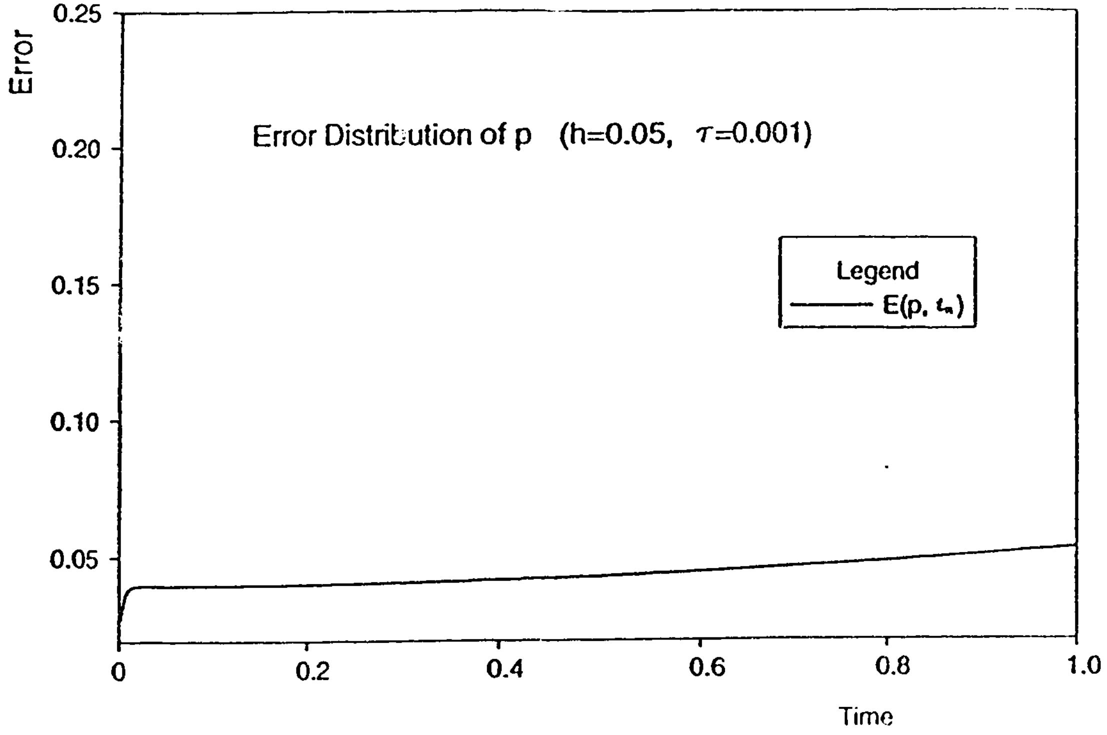


**Figure 6.**

Numerical Solution of u at $t = 0.5$ ( $h = 0.05$, $\tau = 0.001$ )

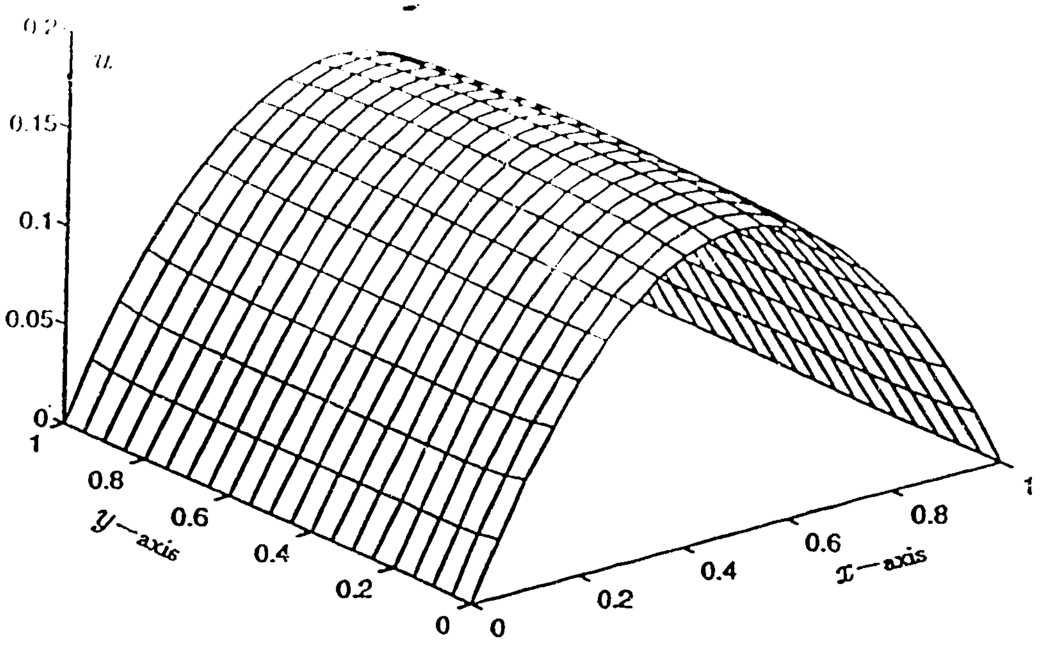


Figure 7.

Numerical Solution of u at $t = 1.0$ ( $h = 0.05$, $\tau = 0.001$ )

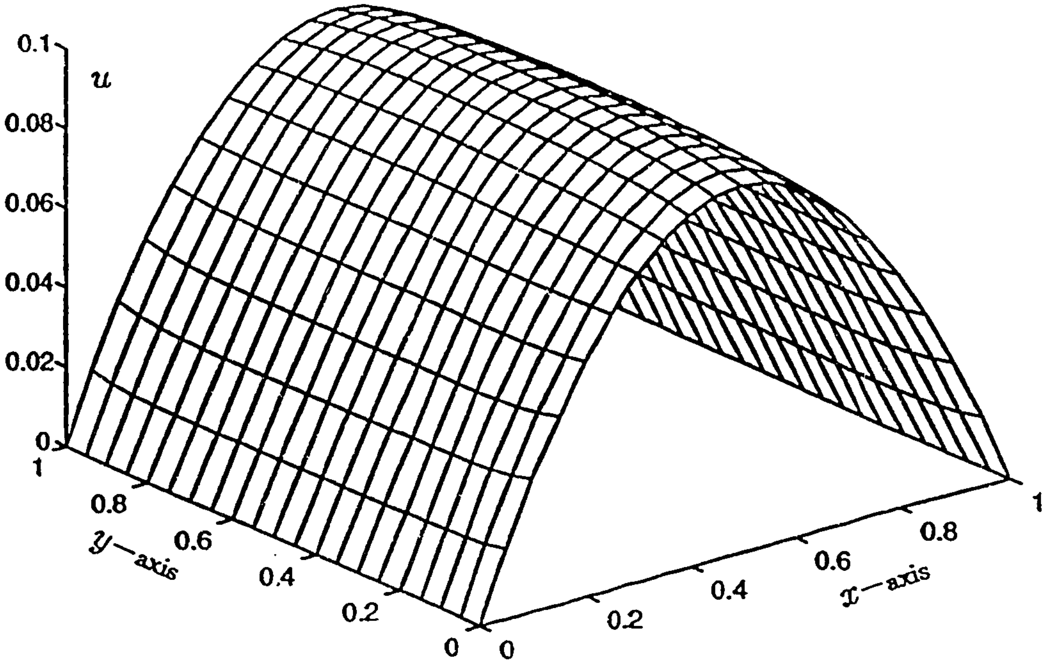


Figure 8.

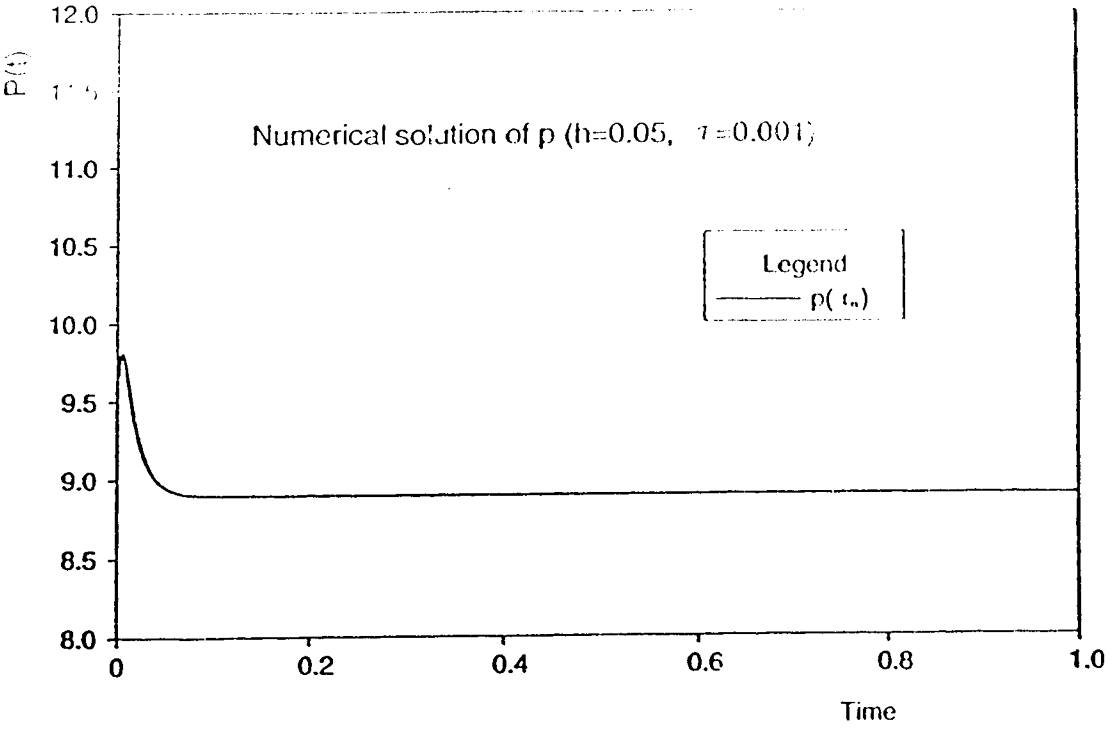


**Figure 9.**

Numerical Solution of u at $t = 0.5$ ( $h = 0.05$, $\tau = 0.001$ )

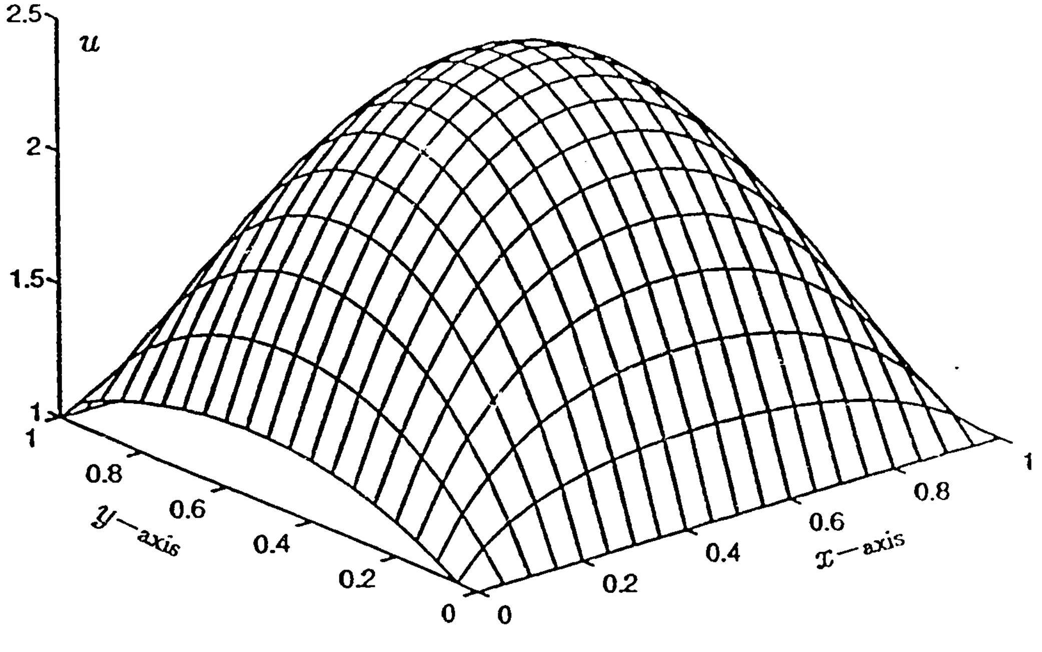


**Figure 10.**

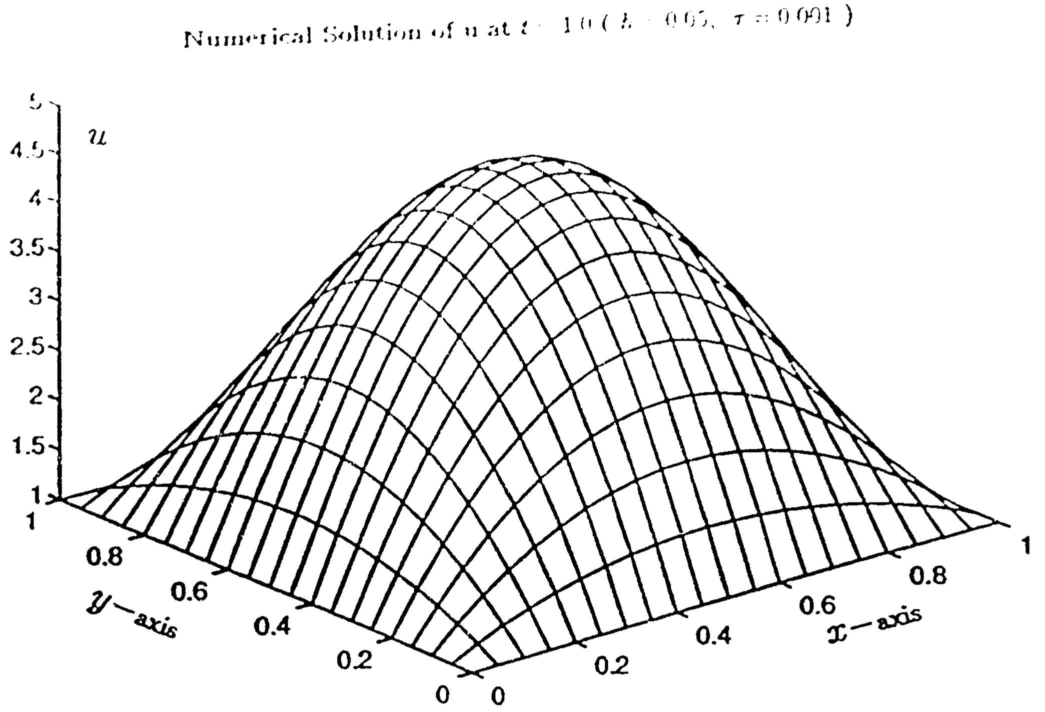


**Figure 11.**

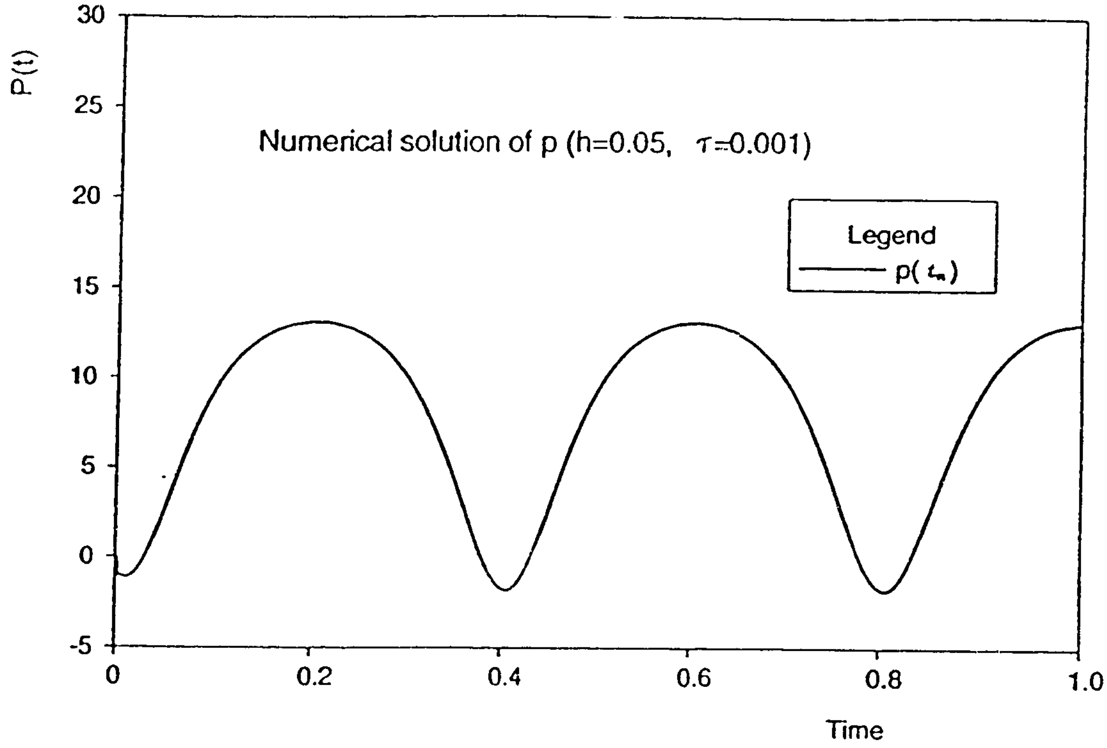


**Figure 12.**

# REFERENCES

1. J. R. Cannon, The one-dimensional heat equation, Encyclopedia of Mathematics and Its Applications, Vol 23, Addison-Wesley, Reading, Massachusetts, 1984.

2. J. R. Cannon, and Y. Lin, Determination of a parameter $p(t)$ in some quasi-linear parabolic differential equations, Inverse Problems, 4 (1988), 35-45.

3. J. R. Cannon, and Y. Lin, Determination of a parameter $p(t)$ in a Hölder class for some semi-linear parabolic equations, Inverse Problems, 4 (1988), 596-605.

4. J. R. Cannon and Y. Lin, An inverse problem of finding a parameter in a semi-linear heat equation, J. Math. Anal. Appl., 2(145), (1990), 470-484.

5. J. R. Cannon, Y. Lin and S. Wang, Determination of source parameter in parabolic equations, MECCANICA, 27(1992), 85-94.

6. J. R. Cannon and H. Yin, On a class of non-classical parabolic problems, J. Diff. Eqs., 2(79), (1989), 266-288.

7. A. Friedman, Partial differential equations of parabolic type, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.

8. O. A. Ladyzenskaja, V. A. Solonnikov and N. N. Uralceva, Linear and quasilinear equations of parabolic type, A.M.S. Tran. Math. Mono., 23, Providence, R.I., 1968.

9. Y. Lin, An inverse problem for a class of quasilinear parabolic equations, SIAM J. Math. Anal., 22(1991), 146-156.

10. A. I. Prilepko and D. G. Orlovskii, Determination of the evolution parameter of an equation and inverse problems of mathematical physics, I, Diff. Eqs., Vol. 1(21), 1985, 119-129, II, 4(21), 1985, 694-701.

11. A. I. Prilepko and V. V. Solo'ev, Solvability of the inverse boundary value problem of finding a coefficient of a lower order term in a parabolic equation, Diff. Eqs., 1(23), 1987, 136-143.

12. S. Wang and Y. Lin, A finite difference solution to an inverse problem for determining a control function in a parabolic partial differential equation, Inverse Problems, 5(1989), 631-640.

The results of this chapter are joint work with J. R. Cannon and Y. Lin and appeared in Inverse Problems, 10(1994), 227-243.

## Chapter 5

## Finite Difference Approximations for a Class of Paraboiic Equations With Non-local Boundary Conditions

### §5.0. Introduction

In this chapter we study finite difference approximations to the solution of the following parabolic equations with non-local boundary condition:

$$
\begin{aligned}
&u_t - \Delta u = 0 \ \text{ in } \ Q_T, \\
&u(x,y,0) = \phi(x,y), \qquad (x,y) \in \Omega, \\
&u(x,y,t) = \int_\Omega K(x,y,\xi,\eta)u(\xi,\eta,t)d\xi d\eta, \qquad \text{on } \ \partial\Omega \times [0,T),
\end{aligned}
$$

where $Q_T = \Omega \times (0,T)$, $T > 0$, $\Omega = (0,1) \times (0,1)$, $\phi(x,y) \not\equiv 0$ and $K(x,y,\xi,\eta)$ are known functions. In addition, it is assumed that for some constant $0 < \rho < 1$ the kernel $K(x,y,\xi,\eta)$ satisfies

$$
\int_\Omega |K(x,y,\xi,\eta)|d\xi d\eta \leq \rho < 1, \qquad \forall (x,y) \in \partial\Omega.
$$

In [4, 5] Day considered the one-dimensional problem on $(-L,L)$, $L > 0$, with the boundary conditions

$$
u(-L,t) = \int_{-L}^{L} f_1(x)u(x,t)dx \qquad \textit{and} \qquad u(L,t) = \int_{-L}^{L} f_2(x)u(x,t)dx,
$$

and showed that if

$$
\int_{-L}^{L} |f_1(x)|dx < 1 \qquad \textit{and} \qquad \int_{-L}^{L} |f_2(x)|dx < 1,
$$

then, for the solution $u$,

$$
U(t) = \max_{-L \leq x \leq L} |u(x,t)|
$$

is decreasing in $t$. The solution $u$ represents the entropy in a quasi-static theory of thermoelasticity [5, 6], so that Day's results show that the maximum modulus of the entropy is decreasing in time. In [8] Friedman extended Day's results to a general parabolic equation in n-dimensions of the form

$$\frac{\partial u}{\partial t} - \sum_{i,j=1}^{n} a_{i,j}(x,t)\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^{n} a_i(x,t)\frac{\partial u}{\partial x_i} + au = 0$$

with $a(x,t) \geq 0$ and with the initial and boundary conditions as given in our original problem. Moreover, Friedman proved that there exists $C_0 > 0$ and $\lambda > 0$ such that

$$U(t) \leq C_0 e^{-\lambda t}, \qquad t \geq 0,$$

i.e., $U(t)$ decays to zero exponentially as $t \to \infty$. Problems similar to the abov also arise from the determination of the unknown source parameter [2, 9] and other related problems [10].

For physical applications of the original problem, let us consider first the coupled partial differential equations

$$a\theta_{xx} = b\theta_t + \theta_0 B v_{xxt}, \qquad A v_{xxxx} = B\theta_{xx}$$

which describe the quasi-static flexure of a thermoelastic rod [5]. Here $\theta(x,t)$ is the temperature, $\theta_0$ is a uniform reference temperature, $v(x,t)$ is the transverse displacement, $a$ is the conductivity, $b$ is the specific heat at the constant strain, the constant $A$ is the flexure rigidity and the constant $B$ is a measure of the cross-coupling between thermal and mechanical effects. We assume that the ends $x = -L$ and $x = L$ to be maintained at the reference temperature $\theta_0$ and to be clamped, that is

$$\theta(-L,t) = \theta(L,t) = 0,$$

$$u(-L,t) = u_x(-L,t) = u(L,t) = u_x(L,t) = 0.$$

Let

$$u = \frac{b}{\theta_0}(\theta - \theta_0) + Bu_{xx}$$

be the entropy. Then after some mathematical manipulations [5] we obtain that $u$ satisfies

$$au_{xx} = (b + \theta_0 \frac{B^2}{A})u_t$$

with the boundary conditions

$$u(-L,t) = -\frac{\theta_0 B^2}{2bA^2L^2}\int_{-L}^{L}(L - 3x)u(x,t)dx,$$
$$u(L,t) = -\frac{\theta_0 B^2}{2bA^2L^2}\int_{-L}^{L}(L + 3x)u(x,t)dx$$

and an appropriate initial condition.

For the second example we consider the equations [1, 3, 5]

$$\begin{aligned} a\theta_{xx} &= b\theta_t + \theta_0\alpha(3\lambda + 2\mu)v_{xt} \\ (\lambda + 2\mu)v_{xx} &= \alpha(3\lambda + 2\mu)\theta_x \end{aligned}$$

which describe the heat conduction behavior of a slab $-L \leq x \leq L$ made of homogeneous and isotropic material. Here $\theta(x,t)$ is the temperature, $v(x,t)$ is the displacement component in the direction of the $x-$axis, $\theta_0$ is a uniform reference temperature, $\alpha$ is the coefficient of expansion, and $\lambda$ , $\mu$ are the elastic moduli. The boundary conditions are

$$\theta(-L,t) = \theta(L,t) = \theta_0, \quad v(-L,t) = v(L,t) = 0.$$

Let

$$u = \frac{b}{\theta_0}(\theta - \theta_0) + \alpha(3\lambda + 2\mu)v_x$$

be the entropy, one has [5] that $u$ satisfies

$$au_{xx} = b^* u_t$$

with the boundary conditions

$$u(-L,t) = u(L,t) = -\frac{b^* - b}{b}\frac{1}{2L}\int_{-L}^{L} u(x,t)dx,$$

where

$$b^* = b + \theta_0\alpha^2\frac{(3\lambda + 2\mu)^2}{\lambda + 2\mu}.$$

For the detailed derivations of the above equations we refer to [1, 3, 5]. The condition

$$\int_\Omega |K(x,y,\xi,\eta)|d\xi d\eta \leq \rho < 1, \qquad \forall (x,y) \in \partial\Omega$$

implies for the first problem that $5\theta_0 B^2 < 3bA^2$ and for the second problem that $\dfrac{b^* - b}{b} < 1$ or $\theta_0\alpha^2(3\lambda - 2\mu)^2 < (\lambda + 2\mu)b$.

Fairweather and López-Marcos studied second-order methods to treat this type of boundary conditions [11][12]. Here we study a less accurate yet simpler finite difference schemes for the original problem. The finite difference procedures proposed below preserve monotonicity, the maximum principle and the exponential decay (if the kernel is non-negative) of the solution for the original equation; therefore, they are considered as good numerical approximations.

Let $h = \Delta x = \Delta y = 1/N$ for some integer $N > 1$, and let $\tau > 0$ be a small step-size in time with $t_n = n\tau$, $n = 0, 1, \cdots$. For a smooth function $v \in C^2(\bar{\Omega})$ we assume that the following numerical integration formula is valid:

$$\int_\Omega K(x,y,\xi,\eta)v(\xi,\eta)d\xi d\eta = \sum_{m,l=0}^{N} w_{m,l}K(x,y,x_m,y_l)\, v_{m,l} + O(h^2),$$

where $w_{m,l} \geq 0$ are weights and $v_{m,l} = v(x_m, y_l)$ with $x_m = m\Delta x$, $y_l = l\Delta y$, $m, l = 0, 1, \cdots, N$. For any $0 < \rho^* < 1$, we restrict $h$ to be so small, say for some $h_0 > 0$, $0 < h \leq h_0$, that

$$\sum_{m,l=0}^{N} w_{m,l}\,|K(x,y,x_m,y_l)| \leq \rho^* = \frac{1+\rho}{2} < 1, \quad (x,y) \in \Omega.$$

Here and throughout this chapter we assume that $h$ is small enough so that this condition is satisfied. In fact this serves as a discrete version of

$$\int_\Omega |K(x,y,\xi,\eta)| d\xi d\eta \leq \rho < 1, \qquad \forall (x,y) \in \partial\Omega.$$

In order to obtain the numerical solution which preserves as many properties of the solution as possible, this condition is necessary and cannot be considered as a constraint on space discretization. For example the weights can be chosen by using trapezoidal rule,

$$w_{m,l} = \begin{cases} \Delta x \Delta y, & m,l = 1,2,\cdots,N-1; \\ \frac{1}{4}\Delta x \Delta y, & m,l \in \{0,N\}; \\ \frac{1}{2}\Delta x \Delta y, & otherwise. \end{cases}$$

Define the following shorthand notations:

$$\partial_t g^n = \frac{g^n - g^{n-1}}{\tau},$$
$$\Delta^2 g_{i,j} = \frac{g_{i+1,j} + g_{i-1,j} + g_{i,j+1} + g_{i,j-1} - 4g_{i,j}}{h^2}.$$

We now define our first numerical scheme: Find $\{U^n_{i,j}\}$ such that

$$\partial_t U^n_{i,j} - \Delta^2 U^n_{i,j} = 0, \qquad i,j = 1,2,\cdots,N-1, \quad n \geq 1,$$
$$U^0_{i,j} = \phi_{i,j}, \qquad i,j = 0,1,2,\cdots,N,$$
$$U^n_{i,j} = K_{i,j}\left(\{U^n_{m,l}\}\right), \quad \{i,j\} \bigcap \{0,N\} \neq \emptyset, \quad n \geq 1,$$

where

$$K_{i,j}\left(\{U^n_{m,l}\}\right) = \sum_{m,l=0}^{N} w_{m,l} K(x_i,y_j,x_m,y_l)\, U^n_{m,l}, \quad \{i,j\} \bigcap \{0,N\} \neq \emptyset, \quad n \geq 1.$$

This fully-implicit scheme requires a full-matrix system to be solved at each time level due to the boundary integration. For small $h > 0$, the matrix will be diagonally-dominant. For the diagonal entries correspond to interior points this is obvious since the diagonal entry is $1+4\frac{\tau}{h^2}$ while there are exactly four nonzero off-diagonal entries of $-\frac{\tau}{h^2}$. For the diagonal entries which correspond to boundary grid points, when $\{i,j\}\bigcap\{0,N\}\neq\emptyset$ and $h$ small enough, we have

$$\sum_{\{m,l\}\neq\{i,j\}} w_{m,l}\,|K(x,y,x_m,y_l)| \leq \frac{1+\rho}{2} < 1-w_{i,j}K(x_i,y_j,x_i,y_j), \quad (x,y)\in\Omega.$$

This tells us the matrix is diagonally-dominant. Thus the resulting system can be solved by Gaussian-elimination or any standard method.

Since the fully implicit scheme results in an error $O(h^2+\tau)$, we may propose the following numerically economical semi-implicit scheme: Find $\{W_{i,j}^n\}$ such that

$$\begin{aligned}
&\partial_t W_{i,j}^n - \Delta^2 W_{i,j}^n = 0, \qquad i,j=1,2,\cdots,N-1, \quad n\geq 1,\\
&W_{i,j}^0 = \phi_{i,j}, \qquad i,j=0,1,2,\cdots,N,\\
&W_{i,j}^n = K_{i,j}\left(\{W_{m,l}^{n-1}\}\right), \quad \{i,j\}\bigcap\{0,N\}\neq\emptyset, \quad n\geq 1.
\end{aligned}$$

We will see that this semi-implicit scheme results in an error $O(h^2+\tau)$ (Theorem 2.2) and is easy to implement numerically since only a pent-diagonal matrix system needs to be solved at each time level. Therefore, it is a very economical and fast algorithm. In addition, it is also unconditionally stable. Alternative methods, say ADI, may also be used to solve the semi-implicit scheme.

### §5.1. Stability, Monotonicity and Exponential Decay

In this section we prove some monotonicity, maximum principle and exponential decay properties for our numerical solutions $U_{i,j}^n$ and $W_{i,j}^n$. Define

$$U^n = \max_{0\leq i,j\leq N}|U_{i,j}^n|, \qquad W^n = \max_{0\leq i,j\leq N}|W_{i,j}^n|.$$

**Theorem 1.1.** *Assume that $U_{i,j}^n$ is a solution of the fully implicit scheme and the* *initial approximation* $U^0_{i,j} \not\equiv 0$ *for* $i, j = 1, 2, \cdots, N-1$. *Then the following holds:*

$$0 < U^n < U^{n-1}, \qquad \forall n \geq 1.$$

**Proof:** We show that

$$0 < U^1 < U^0.$$

We observe that $U^n \geq 0$ for all $n \geq 0$. Consider the first two levels $n = 0$ and $n = 1$. Assume to the contrary that $U^0 \leq U^1$, then $U^1 > 0$. If $U^1 = |U^1_{i_0,j_0}| = U^1_{i_0,j_0} > 0$ ( the case $U^1_{i_0,j_0} < 0$ can be treated in a similar way) for some $(i_0, j_0)$. Then it follows from the discrete maximum principle [7] that this maximum is attained at the boundary. Thus $(i_0, j_0)$ can be selected to be a boundary point. Then, we see from

$$\sum_{m,l=0}^{N} w_{m,l} \left| K(x, y, x_m, y_l) \right| \leq \rho^* = \frac{1+\rho}{2} < 1, \quad (x, y) \in \Omega$$

and our discretization schemes that

$$U^1 = |K_{i_0,j_0}\left(\{U^1_{m,l}\}\right)| \leq \rho^* U^1$$

which is impossible unless $U^1 = 0$ since $\rho^* < 1$. Thus $U^0 > U^1 \geq 0$. By the fully implicit scheme, $U^1_{i,j} = 0$ will lead to $U^0_{i,j} = 0$ for $i, j = 1, 2, \cdots, N-1$ which is a contradiction. Thus, $0 < U^1 < U^0$.

Now, we consider the levels $n = 1$ and $n = 2$. By repeating the above argument with $U^1_{i,j}$ as the initial data, we can show that $0 < U^2 < U^1$. Thus, $0 < U^n < U^{n-1}$ is proved by repeating the above argument for higher levels. □

**Theorem 1.2.** *Assume that* $W^n_{i,j}$ *is a solution of the semi-implicit scheme a* [illegible] *initial approximation* $W^0_{i,j} \not\equiv 0$ *for* $i, j = 1, 2, \cdots, N-1$, *then the following holds:*

$$0 < W^n < W^{n-1}, \qquad \forall n \geq 1.$$

**Proof:** We show first that $W^1 > 0$. If on the contrary $W^1 = 0$, we will get $W_{i,j}^0 = 0$ for $i, j = 1, 2, \cdots, N-1$ by the semi-implicit scheme. This contradiction proves $W^1 > 0$. By

$$\sum_{m,l=0}^{N} w_{m,l} |K(x, y, x_m, y_l)| \leq \rho^* = \frac{1+\rho}{2} < 1, \quad (x, y) \in \Omega,$$

we get

$$|W_{i,j}^1| \leq \rho^* W^0 < W^0, \quad \{i, j\} \bigcap \{0, N\} \neq \emptyset.$$

If $W^1 \geq W^0$, then according to the discrete maximum principle[7],

$$W^1 = \max_{\{i,j\} \bigcap \{0,N\} \neq \emptyset} |W_{i,j}^1| < W^0$$

which is a contradiction. The remainder of the proof follows from an argument similar to the above and mathematical induction. □

**Remark:** Theorem 1.1 and Theorem 1.2 imply the unconditional stability of nun ·- ical solutions $U_{i,j}^n$ and $W_{i,j}^n$, even though $W_{i,j}^n$ is the solution of a semi-implicit finite difference scheme.

In [8] Friedman proved that $U(t)$ decays exponentially when

$$\int_\Omega |K(x, y, \xi, \eta)| d\xi d\eta \leq \rho < 1, \qquad \forall (x, y) \in \partial\Omega$$

is satisfied. We have proved that both $\{U^n\}$ and $\{W^n\}$ possess the strict monotonicity. In fact numerically there exists $\lambda > 0$, as suggested by the examples of the last section, Figure 6 and Figure 7, such that

$$\log \frac{U^{n+1}}{U^n} \sim -\lambda \Delta t \qquad as \quad n \rightarrow \infty,$$

and the same is true for $W^n$. This motivates the justifications of the exponential decay of $U^n$ and $W^n$ when the kernel is non-negative.

**Theorem 1.3.** *Under the assumption that $U^n_{i,j}$ is the solution of the fully implicit scheme, the kernel $K(x,y,\xi,\eta) \geq 0$ and*

$$\int_\Omega |K(x,y,\xi,\eta)| d\xi d\eta \leq \rho < 1, \qquad \forall (x,y) \in \partial\Omega,$$

*there exists a positive constant $\lambda > 0$ such that for $U^0 = \max_{i,j} |U^0_{i,j}|$,*

$$U^n \leq U^0 e^{-\lambda t_n} \quad \textit{for} \quad \textit{all} \quad n \geq 0.$$

**Proof:** Let $V(x,y,t) = e^{-\lambda t}(2U^0 - \frac{\epsilon}{4}(x^2 + y^2))$ where $\epsilon$ and $\lambda$ are two positive constants to be chosen below. It follows easily that there exists $\epsilon_0 > 0$ such that

$$V(x,y,0) = 2U^0 - \frac{\epsilon}{4}(x^2 + y^2) > U^0 \quad \textit{on} \quad \Omega \quad \textit{if} \quad 0 < \epsilon \leq \epsilon_0.$$

Also, since $K_{i,j}(\{1\}) \leq \rho^* < 1$, we find

$$K_{i,j}(\{2U^0\}) \leq \rho^* 2U^0 < 2U^0,$$

and then, there exists a positive constant $\epsilon_1 > 0$ small enough such that for all $0 < \epsilon \leq \epsilon_1$,

$$V^n_{i,j} > K_{i,j}(\{V^n_{m,l}\}), \quad \{i,j\} \bigcap \{0,N\} \neq \emptyset, \quad n \geq 1.$$

Thus, we choose $\epsilon = \min\{\epsilon_0, \epsilon_1\}$. It follows from a simple calculation that

$$\partial_t V^n_{i,j} - \Delta^2 V^n_{i,j} = e^{-\lambda t_{n+1}} \left( \epsilon - \lambda e^{\lambda\xi}(2U^0 - \frac{\epsilon}{4}(x_i^2 + y_j^2)) \right), \qquad i,j = 1,2,\cdots,N-1.$$

where $\xi \in (0,\tau)$. As $\lambda e^{\lambda\tau} \to 0$ when $\lambda \to 0$, we have for some $\lambda_0 = \lambda_0(\epsilon) > 0$ ( or $\lambda_0 = \min\{1/\tau, \epsilon/(2eU^0)\}$) such that for all $0 < \lambda \leq \lambda_0$,

$$\partial_t V^n_{i,j} - \Delta^2 V^n_{i,j} > 0, \qquad i,j = 1,2,\cdots,N-1.$$

Now letting $Z^n_{i,j} = V^n_{i,j} - U^n_{i,j}$ with $\lambda = \lambda_0$ and $\epsilon$ chosen above, we see from the fully implicit scheme and the analysis above that

$$\begin{aligned}
&\partial_t Z^n_{i,j} - \Delta^2 Z^n_{i,j} > 0, \quad i,j = 1,2,\cdots,N-1, \quad n \ge 1,\\
&Z^0_{i,j} > 0, \quad i,j = 0,1,2,\cdots,N,\\
&Z^n_{i,j} > K_{i,j}\left(\{Z^n_{m,l}\}\right), \quad \{i,j\}\bigcap\{0,N\} \ne \emptyset, \quad n \ge 1.
\end{aligned}$$

We now show that $Z^n_{i,j} \ge 0$. Assume that $n_0$ is the first level that $Z^n_{i,j}$ may take the negative values, then we have $(i_0, j_0)$ such that

$$Z^{n_0}_{i_0,j_0} = \min_{i,j} Z^{n_0}_{i,j} < 0.$$

It follows from the discrete maximum principle [7] that $(i_0, j_0)$ must be the boundary point, otherwise $\partial_t Z^{n_0}_{i_0,j_0} - \Delta^2 Z^{n_0}_{i_0,j_0} \le 0$ which is not possible. Thus we have from the positivity of the kernel $K$ that

$$-Z^{n_0}_{i_0,j_0} < K_{i,j}\left(\{-Z^{n_0}_{m,l}\}\right) \le \rho^*(-Z^{n_0}_{i_0,j_0})$$

which implies that $Z^{n_0}_{i_0,j_0} = 0$, a contradiction. Hence, we have proved that $U^n_{i,j} \le V^n_{i,j}$. Using a similar argument by treating $-U^n_{i,j}$ it can be shown that $U^n_{i,j} \ge -V^n_{i,j}$. This completes the proof. □

**Theorem 1.4.** *Assume that $W^n_{i,j}$ is the solution of the semi-implicit scheme and*

$$\int_\Omega |K(x,y,\xi,\eta)| d\xi d\eta \le \rho < 1, \qquad \forall (x,y) \in \partial\Omega.$$

*Then there exists a positive constant $\lambda > 0$ such that $W^0 = \max_{i,j} |W^0_{i,j}|$,*

$$W^n \le W^0 e^{-\lambda t_n} \quad for \quad all \quad n \ge 0.$$

**Proof:** The proof consists of an argument similar to that given in the proof of Theorem 1.3. We therefore only give its outline.

Let $V(x,y,t) = e^{-\lambda t}(2W^0 - \frac{\epsilon}{4}(x^2 + y^2))$ and as before, let $\epsilon_0$ be chosen so small that $V(x,y,0) > W^0$ for $0 < \epsilon \leq \epsilon_0$. Because the numerical integration uses the data on the previous level for the boundary condition, we need to first select a $\lambda_0 > 0$ such that $0 < \lambda \leq \lambda_0$,

$$e^{-\lambda\tau} > \rho^*, \quad i.e. \quad e^{-\lambda\tau} 2W^0 > \rho^* 2W^0.$$

Since $e^{-\lambda\tau} \rightarrow 1$ when $\lambda \rightarrow 0$, the existence of such a $\lambda_0$ is not a problem. With $\epsilon_0$ and $\lambda_0$ chosen as above, we then select $\epsilon_1 > 0$ so small that for $0 < \epsilon \leq \epsilon_1$

$$V_{i,j}^n > K_{i,j}(\{V_{m,l}^{n-1}\}), \quad \{i,j\} \bigcap \{0,N\} \neq \emptyset, \quad n \geq 1.$$

Using $\epsilon = \min\{\epsilon_0, \epsilon_1\}$ and the $\lambda_0$ selected above , we select $\lambda_1 > 0$ such that for $0 < \lambda \leq \lambda_1$,

$$\partial_t V_{i,j}^n - \Delta^2 V_{i,j}^n > 0, \quad i,j = 1,2,\cdots,N-1.$$

We take $\lambda = \min\{\lambda_0, \lambda_1\}$. We omit the remainder of the proof which is the same as that given in the proof of Theorem 1.1 with the $\epsilon$ and $\lambda$ as chosen here. □

## §5.2. Convergence and Error Estimates

In this section we study the convergence and error estimates of the numerical procedures proposed in section 5.0. First, we show the following result.

**Theorem 2.1.** *Assume that in addition to*

$$\int_\Omega |K(x,y,\xi,\eta)| d\xi d\eta \leq \rho < 1, \qquad \forall (x,y) \in \partial\Omega$$

*the kernel $K$ in the original problem satisfies $K \in C^2(\overline{\partial\Omega \times \Omega})$ and*

$$K(x,y,\xi,\eta) \geq 0, \quad \forall \quad (x,y,\xi,\eta) \in \partial\Omega \times \Omega.$$

*If the solution $u$ of the original problem is known a priori to be smooth enough, $u \in C^{4,2}(Q_T)$, then there exists a positive constant $C = C(||u||_{C^{4,2}}, ||K||_{C^2}) > 0$ such that the solution $U^n_{i,j}$ of the fully implicit scheme satisfies*

$$\max_{i,j,n} |U^n_{i,j} - u(x_i, y_j, t_n)| \leq C(h^2 + \tau).$$

**Proof:** Let $e^n_{i,j} = U^n_{i,j} - u(x_i, y_j, t_n)$ for all $i, j, n$. Then we see from

$$\sum_{m,l=0}^{N} w_{m,l} |K(x, y, x_m, y_l)| \leq \rho^* = \frac{1+\rho}{2} < 1, \quad (x, y) \in \Omega$$

and the fully implicit scheme that $e^n_{i,j}$ satisfies

$$\begin{aligned}
&\partial_t e^n_{i,j} - \Delta^2 e^n_{i,j} = \tau^n_{i,j}, \qquad i, j = 1, 2, \cdots, N-1, \quad n \geq 1, \\
&e^0_{i,j} = 0, \qquad i, j = 0, 1, 2, \cdots, N, \\
&e^n_{i,j} = K_{i,j}\left(\{e^n_{m,l}\}\right) + \epsilon^n_{i,j}, \quad \{i, j\} \bigcap \{0, N\} \neq \emptyset, \quad n \geq 1.
\end{aligned}$$

Here $\tau^n_{i,j}$ and $\epsilon^n_{i,j}$ are the truncation errors induced by the discretization of the differential equation and numerical integration respectively. Then there exists $L_0 > 0$ such that

$$\max_{i,j,n} |\tau^n_{i,j}| \leq L_0(h^2 + \tau), \qquad \max_{i,j,n} |\epsilon^n_{i,j}| \leq L_0(h^2 + \tau).$$

We now define an auxiliary function $\theta(x, y)$ by

$$\theta(x, y) = \frac{1 - x^2 - y^2}{4} L_0(h^2 + \tau).$$

It is then easy to verify that

$$-\Delta^2 \theta_{i,j} = L_0(h^2 + \tau) \qquad \textit{and} \quad 0 \leq \theta_{i,j} \leq \frac{L_0}{2}(h^2 + \tau).$$

Let $Z^n_{i,j} = \epsilon^n_{i,j} - \theta_{i,j}$ for all $i$, $j$, $n$. We find from these results and the finite difference scheme about $\epsilon^n_{i,j}$ that

$$\begin{aligned}
&\partial_t Z^n_{i,j} - \Delta^2 Z^n_{i,j} \le 0, \quad i,j = 1,2,\cdots,N-1, \quad n \ge 1,\\
&Z^0_{i,j} = -\theta_{i,j} \le 0, \quad i,j = 0,1,2,\cdots,N,\\
&Z^n_{i,j} = K_{i,j}\left(\{Z^n_{m,l}\}\right) + K_{i,j}\left(\{\theta_{m,l}\}\right) - \theta_{i,j} + \epsilon^n_{i,j}, \quad \{i,j\}\bigcap\{0,N\} \ne \emptyset, \quad n \ge 1.
\end{aligned}$$

We now show that there exists $C > 0$ such that $Z^n_{i,j} \le C(h^2+\tau)$ for all $i, j, n$. If $Z^n_{i,j}$ has a positive maximum, then according to the discrete maximum principle it must be attained at a boundary point. Assume that $M = Z^{n_0}_{i_0,j_0} > 0$ with $n_0 \ge 1$, is the positive maximum. From the boundary condition in the previous inequalities we see that

$$\begin{aligned}
M &\le \rho^* M + (\rho^*+1)\max_{i,j}|\theta_{i,j}| + \max_{i,j,n}|\epsilon^n_{i,j}|\\
&\le \rho^* M + \frac{\rho^*+3}{2}L_0(h^2+\tau),
\end{aligned}$$

which implies

$$M \le \frac{\rho^*+3}{2(1-\rho^*)}L_0(h^2+\tau).$$

Hence, we have proved that

$$\epsilon_{i,j} \le \theta_{i,j} + \frac{\rho^*+3}{2(1-\rho^*)}L_0(h^2+\tau).$$

If instead $Z^n_{i,j} = \epsilon^n_{i,j} + \theta_{i,j}$, then a similar argument gives

$$\epsilon_{i,j} \ge -\theta_{i,j} + \frac{\rho^*+3}{2(1-\rho^*)}L_0(h^2+\tau).$$

Therefore, we have

$$\begin{aligned}
|\epsilon_{i,j}| &\le |\theta_{i,j}| + \frac{\rho^*+3}{2(1-\rho^*)}L_0(h^2+\tau).\\
&\le \frac{3L_0}{1-\rho^*}(h^2+\tau),
\end{aligned}$$

which is the result we want. The proof is complete. □

**Theorem 2.2.** *Under the same assumptions of Theorem 2.1, let $W_{i,j}^n$ be the solution of the semi-implicit scheme. Then for some positive constant $C > 0$, independent of $h$ and $\tau$, we have*

$$\max_{i,j,n} |W_{i,j}^n - u(x_i, y_j, t_n)| \le C(h^2 + \tau).$$

**Proof:** The proof follows by a similar argument to that given in the proof of Theorem 2.1. □

**Remark:** The error estimates in Theorem 2.1 and Theorem 2.2 are uniform for all $0 \le t < T$ with $0 < T \le \infty$. This is guaranteed by the condition

$$\int_\Omega |K(x, y, \xi, \eta)| d\xi d\eta \le \rho < 1, \qquad \forall (x, y) \in \partial\Omega.$$

### §5.3. General Smooth Kernel $K(x, y, \xi, \eta)$

In this section we consider the effect on the original problem when the kernel condition

$$\int_\Omega |K(x, y, \xi, \eta)| d\xi d\eta \le \rho < 1, \qquad \forall (x, y) \in \partial\Omega$$

is replaced by:

$$0 \le K(x, y, \xi, \eta) \le K_0, \quad \forall (x, y, \xi, \eta) \in \partial\Omega \times \Omega.$$

In general if the condition

$$\int_\Omega |K(x, y, \xi, \eta)| d\xi d\eta \le \rho < 1, \qquad \forall (x, y) \in \partial\Omega$$

is not satisfied, then the numerical procedure of the fully implicit and the semi-implicit schemes may not be stable uniformly for $0 < t < \infty$. This will be demonstrated both theoretically and through numerical examples below. For these kernels, the stability will depend upon $K_0$ and $T > 0$. Here we consider a class of kernels which satisfy $0 \le K(x,y,\xi,\eta) \le K_0$ but not

$$\int_\Omega |K(x,y,\xi,\eta)|d\xi d\eta \le \rho < 1, \qquad \forall (x,y) \in \partial\Omega.$$

We first consider the continuous problem. Let $w(x,y)$ be an auxiliary function

$$1 \le w(x,y) = 1 + M\left((x-1/2)^d + (y-1/2)^d\right), \qquad d > 0,$$

where $M$ and $d$ (even) are two positive constants to be chosen. Clearly, we have

$$\min_{(x,y)\in\partial\Omega} w(x,y) = 1 + 2M\left(\frac{1}{2}\right)^d$$

Let $u(x,y,t)$ be a solution of the original problem with $K$ satisfying $0 \le K(x,y,\xi,\eta) \le K_0$. Let $v(x,y,t) = \frac{u(x,y,t)}{w(x,y)}$ and find that $v$ satisfies

$$\begin{aligned} v_t &= \Delta v + 2\frac{\nabla w \cdot \nabla v}{w} + \frac{\Delta w}{w} v, \qquad \text{in } Q_T, \\ v(x,y,0) &= \frac{\phi(x,y,t)}{w(x,y)} \qquad (x,y) \in \Omega \\ v(x,y,t) &= \int_\Omega R(x,y,\xi,\eta) v(\xi,\eta) d\xi d\eta, \quad (x,t) \in \partial\Omega, \quad t \ge 0, \end{aligned}$$

where

$$R(x,y,\xi,\eta) = K(x,y,\xi,\eta)\frac{w(\xi,\eta)}{w(x,y)}.$$

Thus, we have from $0 \le K(x,y,\xi,\eta) \le K_0$ that

$$\int_\Omega |R(x,y,\xi,\eta)|d\xi d\eta \le \frac{K_0}{1 + M(1/2)^{d-1}} \int_\Omega w(\xi,\eta) d\xi d\eta.$$

A simple calculation shows that if $d$ is an even integer,

$$\int_\Omega w(\xi,\eta) d\xi d\eta = 1 + M\frac{(1/2)^{d-1}}{d+1}.$$

Then it follows that

$$\int_\Omega |R(x,y,\xi,\eta)|d\xi d\eta \le K_0 \frac{1+M\frac{(1/2)^{d-1}}{d+1}}{1+M(1/2)^{d-1}} \to \frac{K_0}{d+1} \qquad as \qquad M \to \infty.$$

Hence, taking $d = 2K_0$ and $M = M(K_0) > 0$ large enough, we can achieve

$$\int_\Omega |R(x,y,\xi,\eta)|d\xi d\eta \le \frac{K_0}{d+1} \le \frac{2K_0}{2K_0+1} < 1, \quad \forall \quad (x,y) \in \partial\Omega.$$

For $w(x,y)$ chosen in this way, we have for some $K_1 = K_1(K_0) > 0$ that $|\Delta w/w| \le K_1$.

Now consider the transformation $v(x,y,t) = e^{\lambda t}Y(x,y,t)$ with $\lambda \ge K_1$. We find that $Y$ satisfies

$$\begin{aligned} Y_t &= \Delta Y + 2\frac{\nabla w \cdot \nabla Y}{w} + \left(\frac{\Delta w}{w} - \lambda\right) W \qquad \text{in } Q_T, \\ Y(x,y,0) &= \frac{\phi(x,y,t)}{w(x,y)}, \quad (x,y) \in \Omega \\ Y(x,y,t) &= \int_\Omega R(x,y,\xi,\eta)Y(\xi,\eta)d\xi d\eta, \quad (x,t) \in \partial\Omega, \quad t \ge 0. \end{aligned}$$

**Remark:** We now see from [5] that $Y$ obeys the maximum principle and possesses the monotonicity and exponential decay properties, which in turn results in monotonic and stable numerical schemes if the above equation for $Y$ is discretized as the fully implicit scheme or the semi-implicit scheme we proposed.

Turning to numerical approximations for the original problem with the new kernel condition, we let $\tau = T/N_1$ where $N_1$ is a positive integer. Numerical solutions to the problem, $U_{i,j}^n$ or $W_{i,j}^n$, are defined as in the fully implicit scheme or the semi-implicit scheme. We cannot expect that these two schemes have the monotonicity properties as described in Theorem 1.1 and Theorem 1.2 when

$$\int_\Omega |K(x,y,\xi,\eta)|d\xi d\eta \le \rho < 1, \qquad \forall (x,y) \in \partial\Omega$$

is not satisfied. However, we have the following local stability estimates.

**Theorem 3.1.** *Assume that $U^n_{i,j}$ is defined as in the fully implicit scheme or the semi-implicit scheme for the original problem with $0 \le K(x,y,\xi,\eta) \le K_0$. If the solution $u$ of the original problem is known a priori to be smooth enough, $u \in C^{4,2}(Q_T)$, then there is some constant $C^* = C^*(||u||_{C^{4,2}}, ||K||_{C^2}, K_0, T) > 0$ such that*

$$\max_{i,j,n} |U^n_{i,j} - u(x_i, y_j, t_n)| \le C^*(h^2 + \tau).$$

**Proof:** The proof is similar to that given in section 5.2, so is outlined as below. For this local convergence, we let $U^n_{i,j} = e^{\lambda t_n} w_{i,j} Y^n_{i,j}$, where $\lambda$ and $w(x,y)$ are defined as above. Thus, it follows from a simple calculation that $Y^n_{i,j}$ satisfies a difference equation which is the discrete version of the equation for $Y(x,y,t)$. Thus it follows from Theorem 2.1 and Theorem 2.2 (The proof needs only minor modifications from that given in Section 5.2, we therefore omit) that there exists a positive constant $C > 0$ such that

$$\max_{i,j,n} |Y^n_{i,j} - Y(x_i, y_j, t_n)| \le C(h^2 + \tau),$$

where $C$ is independent of $K_0$ and $T > 0$, and then, we obtain that

$$|U^n_{i,j} - u(x_i, y_j, t_n)| \le e^{\lambda t_n} w_{i,j} |U^n_{i,j} - u(x_i, y_j, t_n)| \le C^*(h^2 + \tau),$$

which completes the proof. □

**Remark:** The constant $C^*$ above can be very large if $K_0$ and $T > 0$ are very large. This can be seen from the choices of $d$ and $K_1$ in the above analysis, and also is demonstrated in the examples in the last section. In other words although $h$ and $\tau$ are small, the error could be very big, even approaching $\infty$ as $n \to \infty$.

## §5.4. Numerical examples

We shall report several numerical examples which support our theoretical justifications in the previous sections, i.e., stability, monotonicity and exponential decay as $t \to \infty$. Both semi-implicit and fully explicit schemes using the trapezoidal rule for numerical integration are used in our computations.

**Example 4.1.** In order to demonstrate the error analysis and stability, we select $\Omega = [0, 2\pi] \times [0, 2\pi]$, $\phi(x, y) = \sin(x)\sin(y)$ and $K(x, y, \xi, \eta) = \frac{k}{4\pi}$. Thus, for any real constant $k > 0$, $u(x, y, t) = \sin(x)\sin(y)e^{-2t}$ is the solution with $\int_\Omega |K(x, y, \xi, \eta)| d\xi\eta = k$. Figure 1 and Figure 2 show by using the semi-impiicit scheme that the error distributions of $u$ ( the maximum error on each level via the time) with parameter $k$ varying from 0.1 to 4. Clearly, for $k = 0.1$, 0.3, 0.5 and $k = 0.8$, even $k = 1.0$, the errors are under control as predicted by Theorem 2.1. On the other hand, for $k = 1.5$, 2.5, 3 and $k = 4$, it is seen that the errors are under control only for a short period of time, and then divergent to $\infty$ as $n \to \infty$. This is the exact same result as predicted by Theorem 3.1, i.e., the numerical schemes are stable locally depending upon $K_0 > 0$ and $T > 0$. Figure 3 shows the error distribution of $u$ by using the fully implicit scheme. For $0 < k < 1$ the error distributions of $u$ in this example are almost identical to the case of $k = 1$. Also we noticed that the fully implicit scheme is more stable than the semi-implicit scheme.

**Example 4.2.** We now take a simple model problem with the same spatial domain and kernel as in example 1, $\phi(x, y) = \sin(xy)$ and $k = 0.8$. Figure 4 and Figure 5, by using semi-implicit and fully implicit schemes respectively, show the distribution of $U^n$ via the time $t$, which decreases to zero exponentially as $t \to \infty$. If we assume roughly that for some $\lambda(t)$, $C(t)$ we have

$$U(t) \sim C(t)e^{\lambda(t)t} \quad as \quad t \to \infty,$$

then $\lambda(t)$ can be calculated by the formula

$$\lambda^n \sim \frac{1}{\Delta t} log(\frac{U^{n+1}}{U^n}) \quad as \quad t \to \infty,$$

Figure 6 and Figure 7, by using semi-implicit and fully implicit schemes respectively, show the distributions of $\lambda(t)$ proposed above, and it is seen that $\lambda^n$ approaches a negative constant as expected. For the semi-explicit scheme we find $\lambda^n \sim -0.145$, and the fully explicit $\lambda^n \sim -0.1336$, thus the difference is $1.2 \times 10^{-2}$ which is within the rate of the truncation error of the discretization.

With $\lambda^n$ calculated above we then can compute $C(t)$ by

$$C^n \sim U^n e^{-\lambda^n \Delta t\, n} \quad as \quad n \to \infty.$$

Figure 8 shows the distribution of $C(t)$ computed by the semi-implicit scheme according to the above assumption. In this example we see that $C(t)$ also approaches a constant. Figure 9 and Figure 10 show the numerical solutions of $u$ at $t = 0.5$ and $t = 1.0$ with $h = \pi/20$ and $\tau = 0.01$.

**Example 4.3.** Taking the same model problem as in example 2 except that the initial data is $\phi(x, y) = (\pi - x)(\pi - y)$ and $k = 0.4$. Figure 11, Figure 12 and Figure 13 show the distributions of $U(t)$, $\lambda(t)$ and $C(t)$ using the semi-implicit scheme . It is noticed that $U(t)$ goes exponentially to zero very rapidly as $t \to \infty$ compared to that in example 2. This is due to the fact that $C(t)$ also approaches zero, not a fixed constant as in example 2.

**Remark:** From these examples we have a rough idea how $U(t)$ will behave as the time advances, i.e., we can at least to estimate the parameter $\lambda$ mentioned in Section 5.1 by using our numerical semi-implicit or fully implicit schemes.

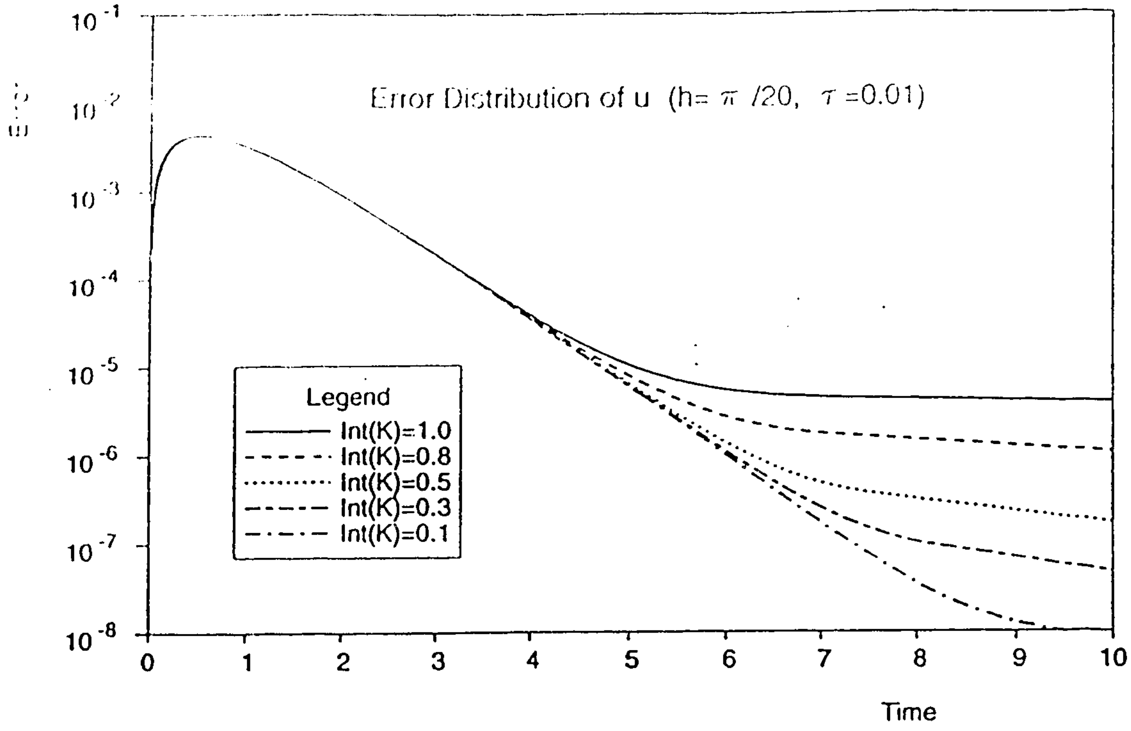


**Figure 1: The semi-implicit scheme**

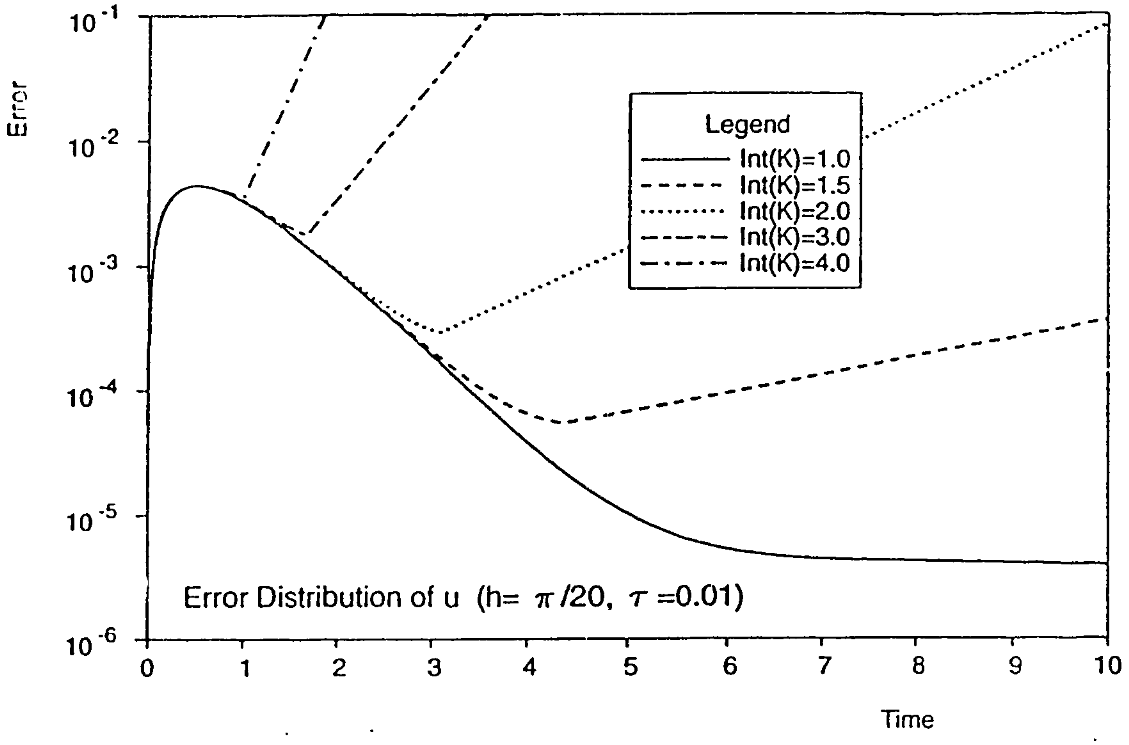


**Figure 2: The semi-implicit scheme**

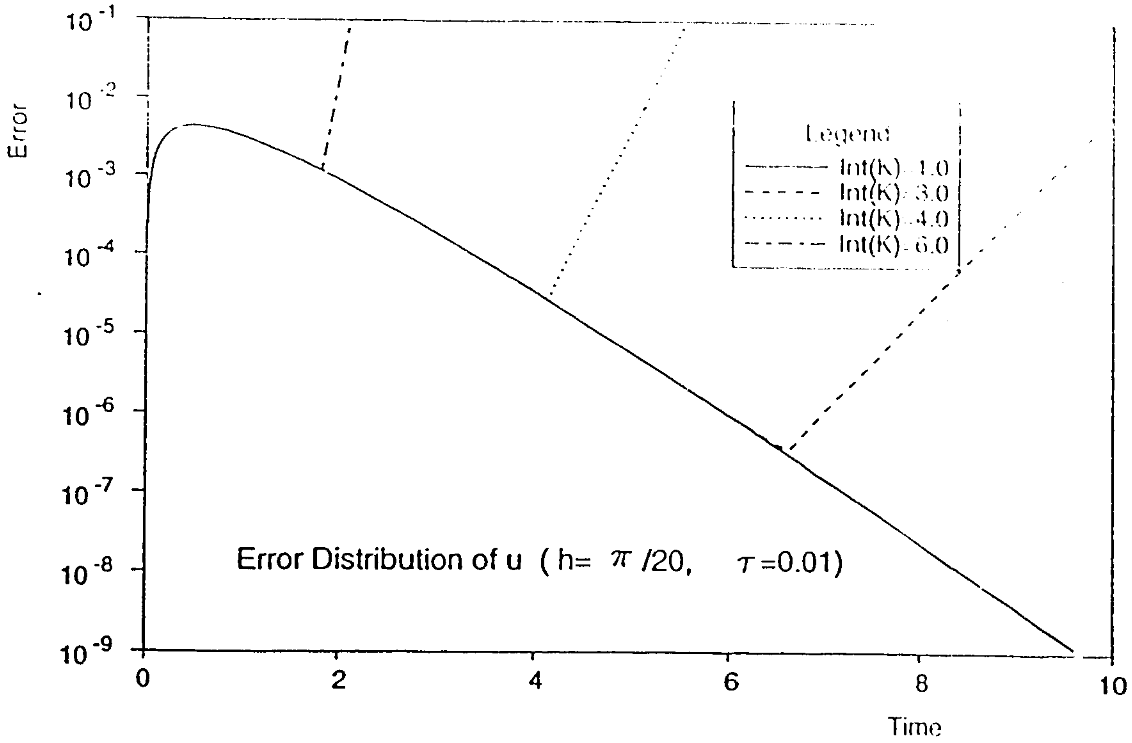


**Figure 3: The fully-implicit scheme**

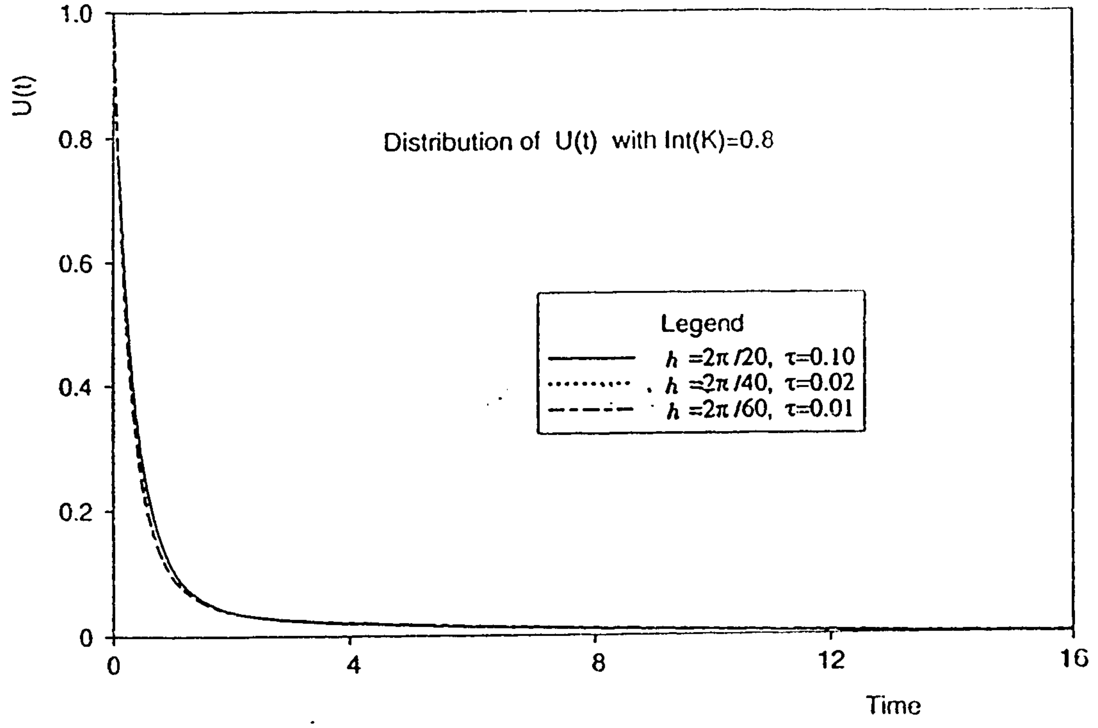


**Figure 4: The semi-implicit scheme**

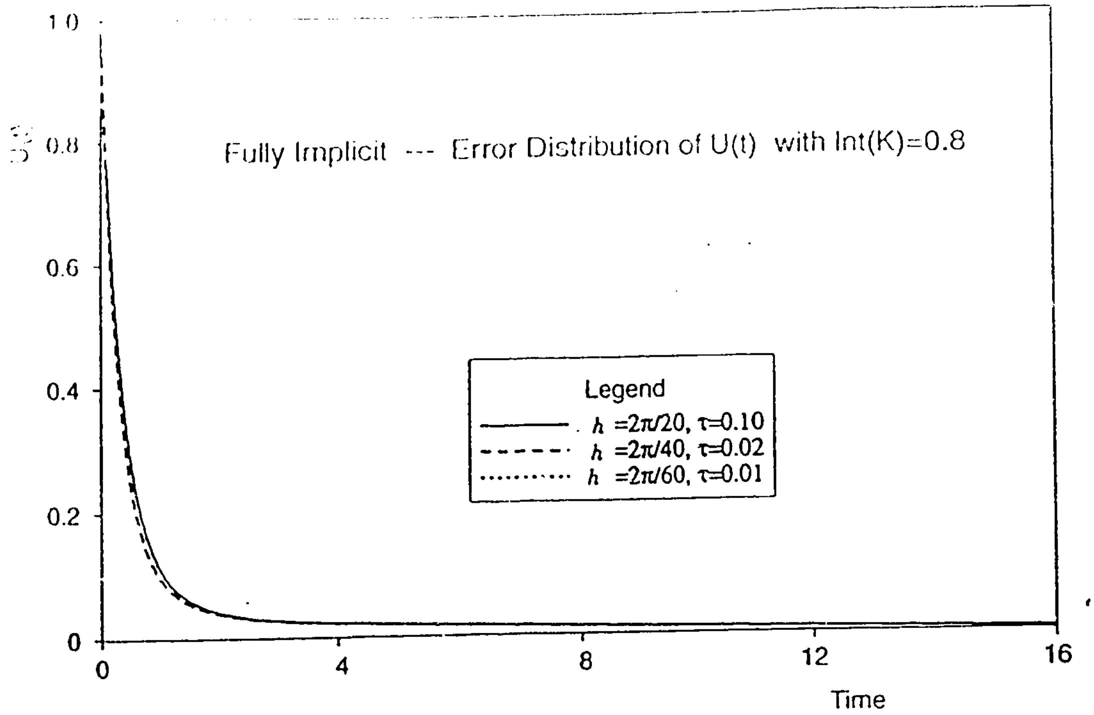


**Figure 5: The fully-implicit scheme**

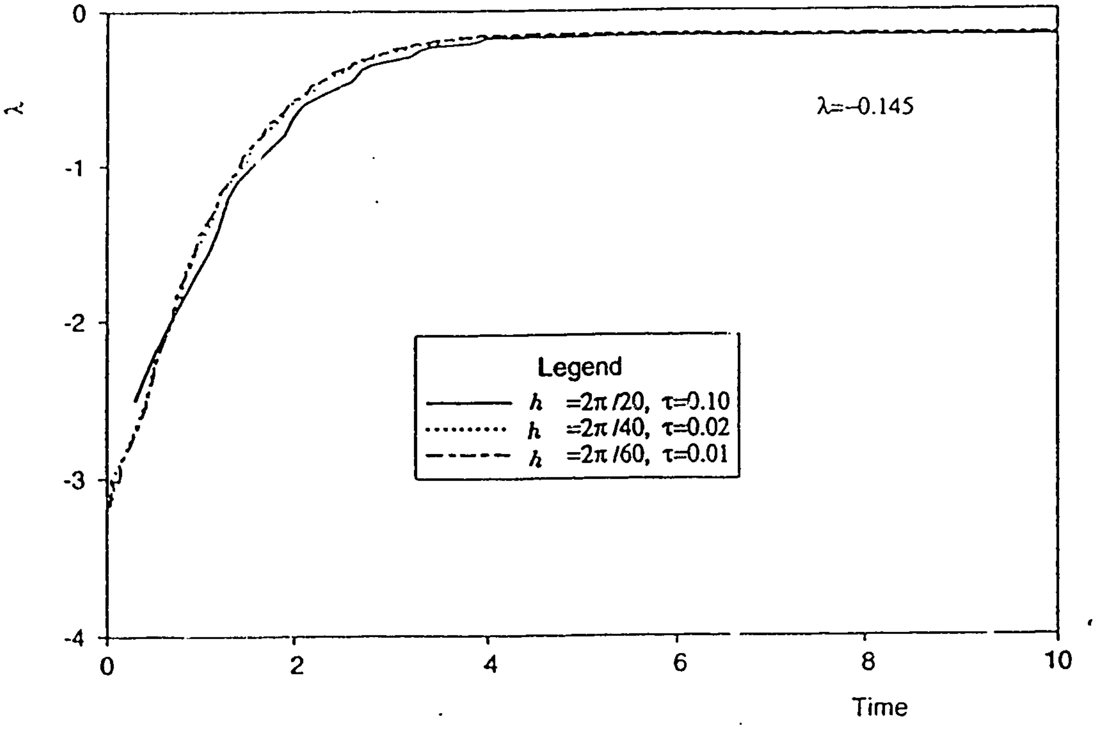


**Figure 6: The semi-implicit scheme**

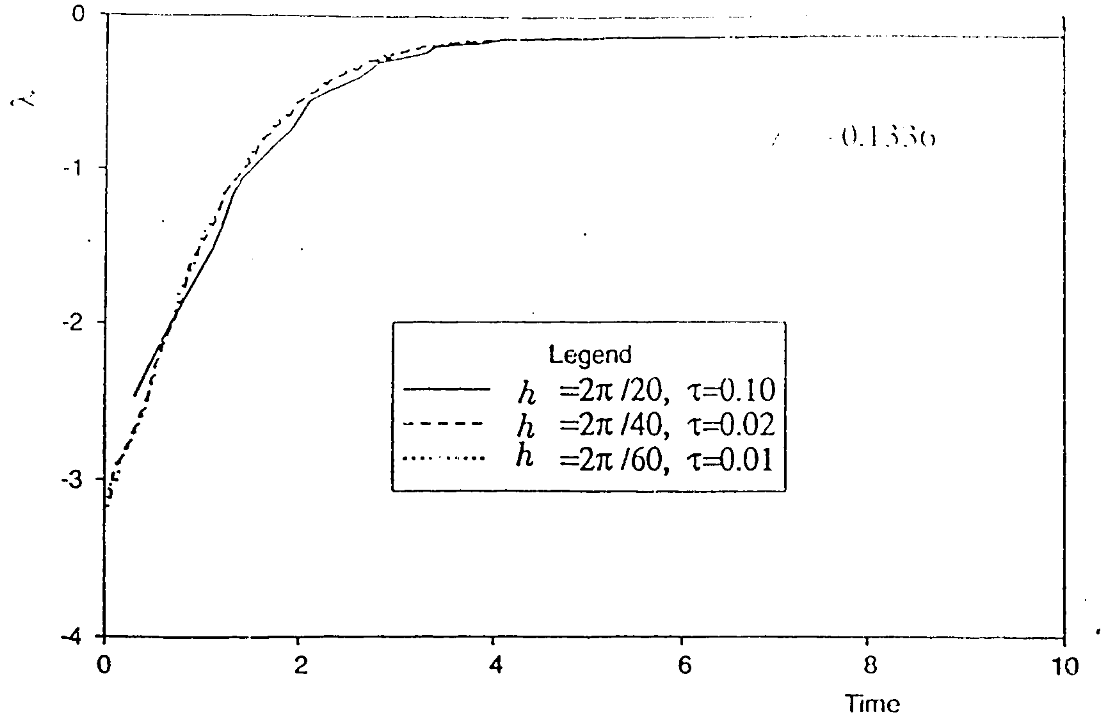


Figure 7: The fully-implicit scheme

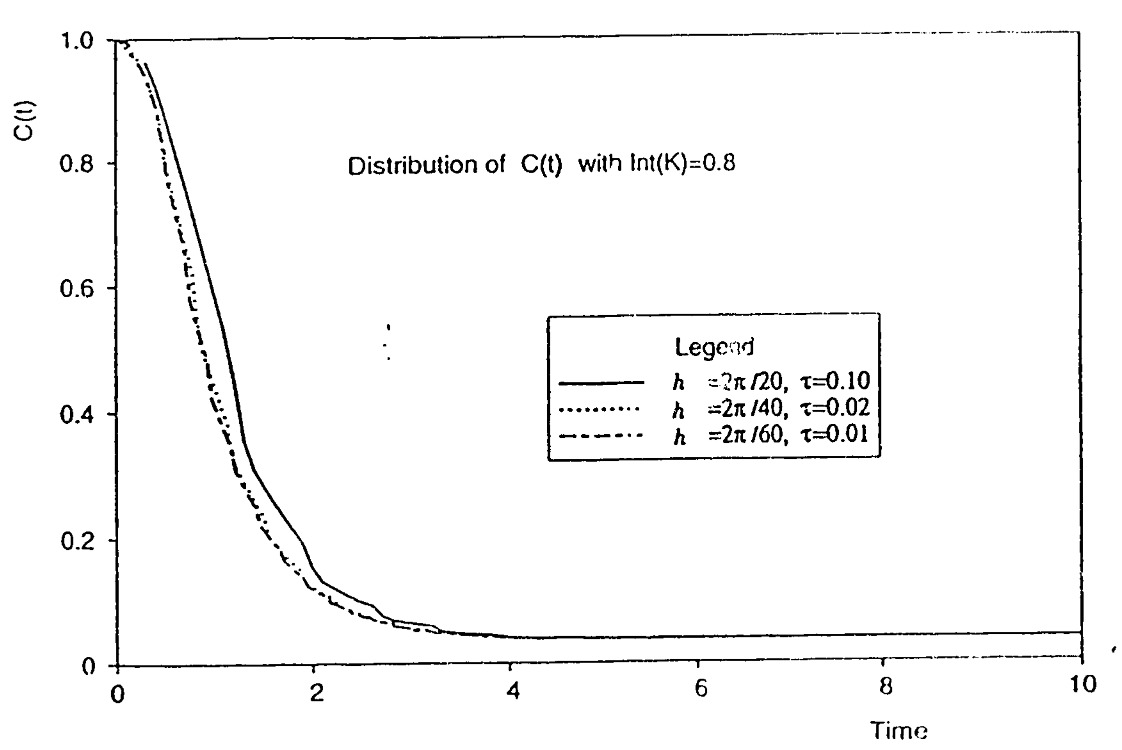


Figure 8: The semi-implicit scheme

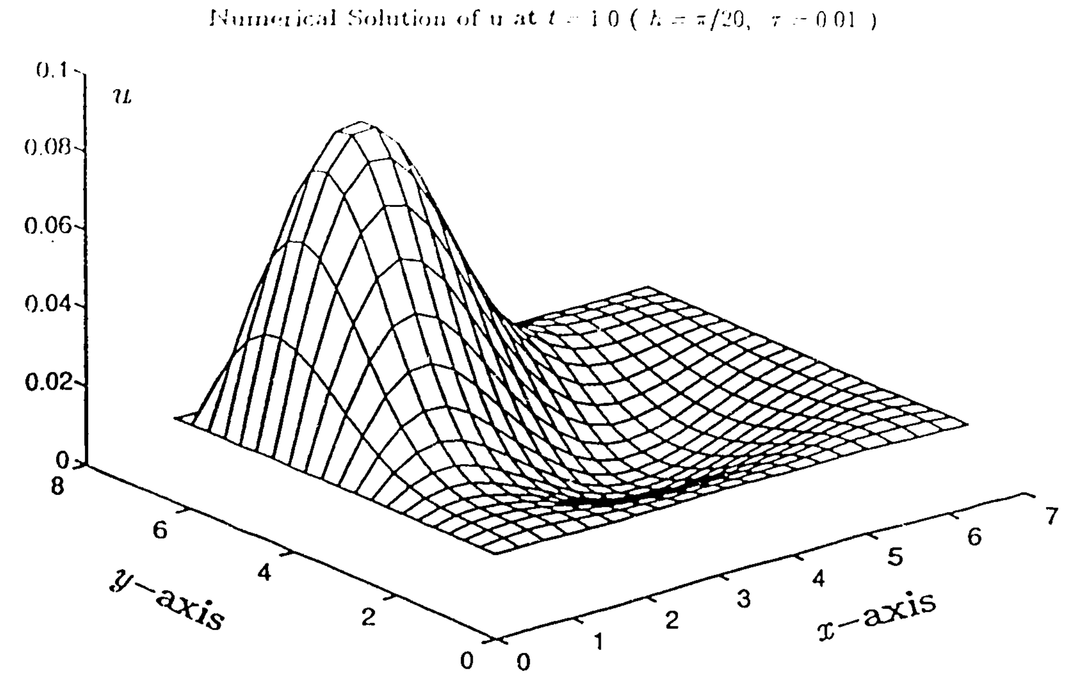


Figure 9: The semi-implicit scheme

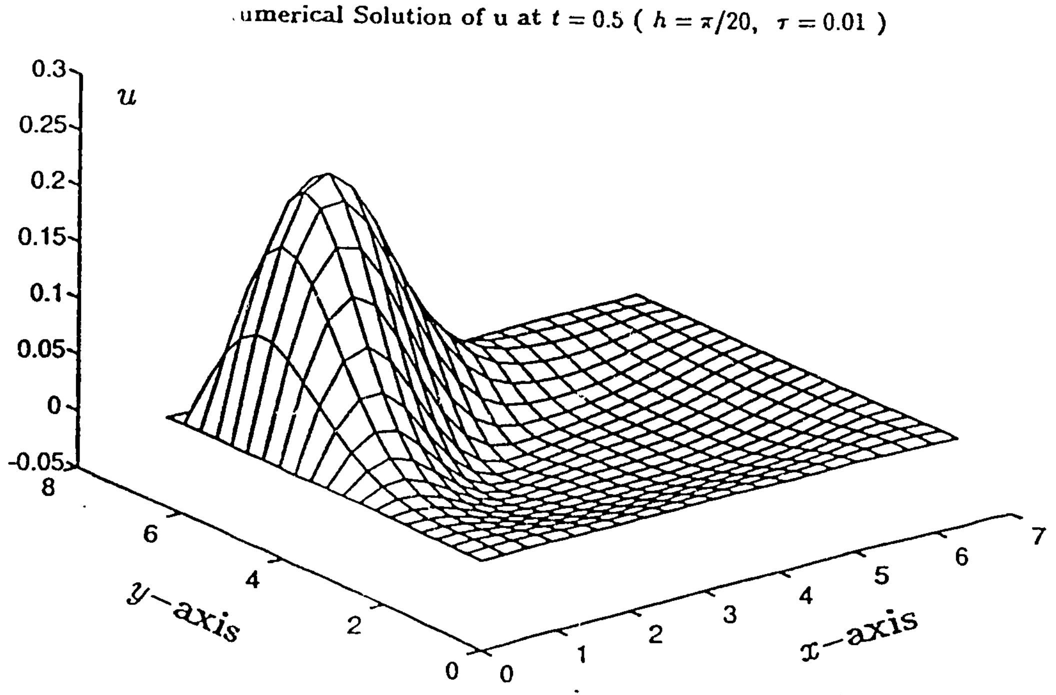


Figure 10: The semi-implicit scheme

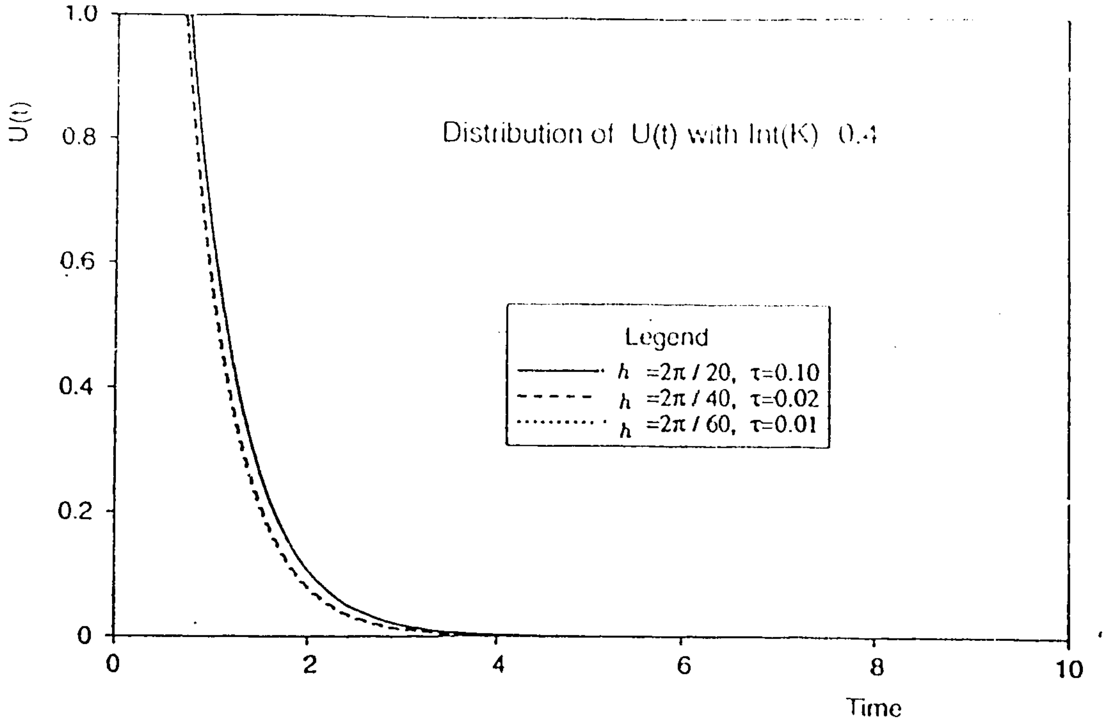


**Figure 11: The semi-implicit scheme**

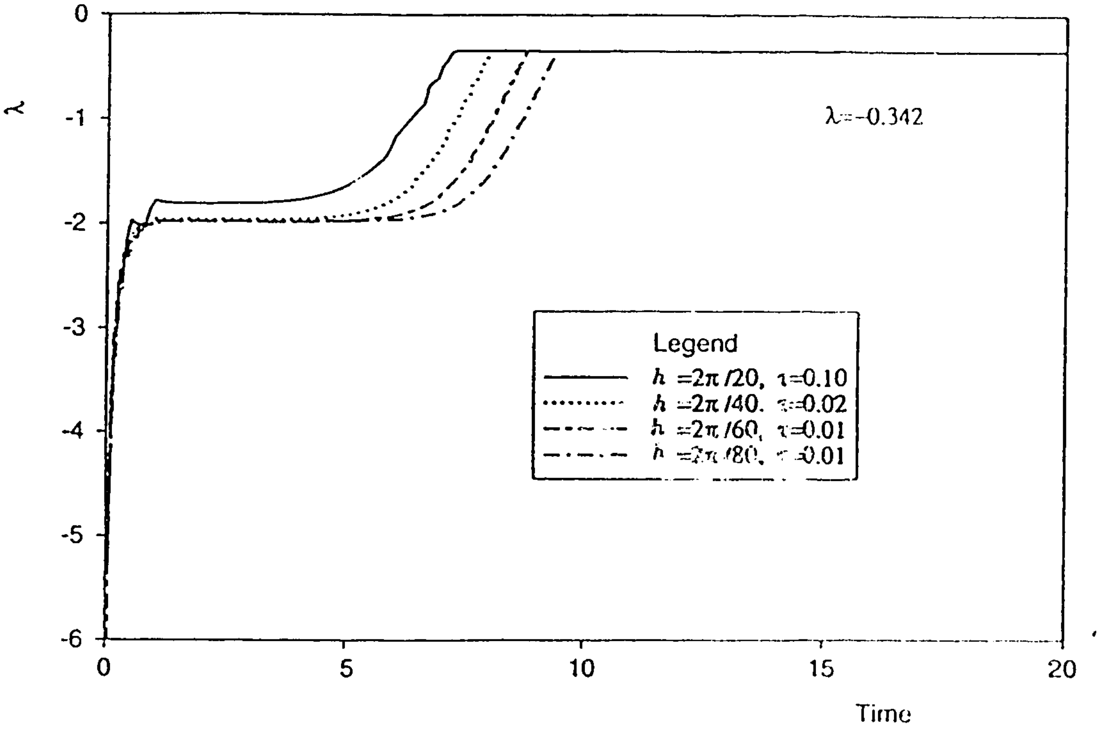


**Figure 12: The semi-implicit scheme**

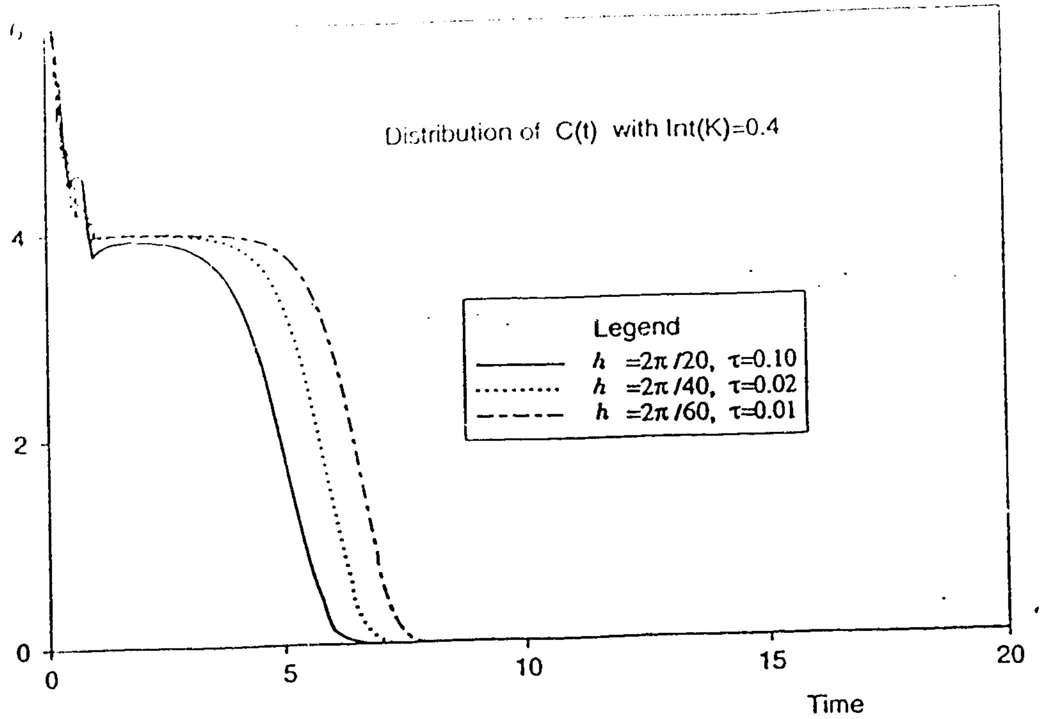


Figure 13: The semi-implicit scheme

## REFERENCES

1. B. A. Boley and J. H. Weiner, Theory of thermal stresses, Wiley, New York, 1960.

2. J. R. Cannon, The one-dimensional heat equation, Encyclopedia of Mathematics and Its Applications, Vol 23, Addison-Wesley, Reading, Massachusetts, 1984.

3. D. E. Carlson, Linear thermoelasticity, Encyclopedia of Physics, VIa/2, Springer, Berlin, 1972.

4. W. A. Day, Existence of a property of solutions of the heat equation to linear thermoelasticity and other theories, Q. Appl. Math., 40 (1982), 319-330.

5. W. A. Day, A decreasing property of solution of a parabolic equation with applications to thermoelasticity, Q. Appl. Math., 41 (1983), 468-475.

6. W. A. Day, Heat conduction within linear thermoelasticity, Springer-Verlag, New York, 1985.

7. J. Douglas, Jr., On the numerical integration of quasi-linear parabolic equations, Pacific J. Math, 6(1956), 35-42.

8. A. Friedman, Monotonic decay of solution of parabolic equations with non-local boundary conditions, Q. Appl. math., XLIV(1986), 401-407.

9. Y. Lin, An inverse problem for a class of quasilinear parabolic equations, SIAM J. Math. Anal., 22(1991), 146-156.

10. N. I. Yurchuk, Mixed problem with an integral condition for certain parabolic equations, Diff. Eqs., 22(1986), 2117-2126.

## ADDITIONAL REFERENCES

11. G. Fairweather and J. C. López-Marcos, A box method for a nonlinear equation of population dynamics, IAM J. Numer. Anal., 11(1991), 525-538.

12. G. Fairweather and J. C. López-Marcos, An implicit extrapolated scheme for the Gurtin-MacCamy equation, Comput. Math. Appl., 27(1994), 41-53.

The results of this chapter are joint work with Y. Lin and H. M. Yin and to appear in International Journal of Mathematics and Mathematical Sciences.