Online Lecture Note

Introduction to Mathematical Modeling and Computation

Version 1.0

April 2, 2018

Satoru Yamamoto

Professor, Dr. Eng.

Laboratory of Mathematical Modeling and Computation

Dept. of Computer and Mathematical Sciences

Tohoku University

Sendai 980-8579, Japan

Email:yamamoto@caero.mech.tohoku.ac.jp

CONTENTS

1. Introduction

2. Entrance of Mathematical Modeling and Computation

2-1 Reaction Equation
2-2 System of Reaction Equations
2-3 Partial Differential Equation
2-4 Diffusion Equation
2-5 Types of PDE
2-6 Finite Difference of Laplace Equation
2-7 Taylor’s Expansion
2-8 Boundary Condition

3. FDM for Mathematical Models based on PDE

3-1 Relaxation Method for Laplace Equation
3-2 3D Potential Flow solved by SOR Method
3-3 FDM for Equation of Heat Conduction
3-4 Linear Stability Analysis
3-5 Direct Method
3-6 Relaxation Method for Equation of Heat Conduction
3-7 Excel Macro Program for 2D Equation of Heat Conduction
3-8 Application to 3-D Heat Conduction over A Cubic Body
3-9 Characteristic Equation

4. Mathematical Modeling based on Reaction Equations

4-1 SIR model and the Extension
4-2 SEIR model

5. Mathematical Modeling based on Reaction-Diffusion Equations

5-1 Reaction-Diffusion Equation
5-2 System of RDE’s
5-3 A Mathematical Model for Bacteria and White Cell
5-4 Mathematical Model for Self-replicating Pattern

6. Concluding Remarks

References

1. Introduction

Mathematical Modeling and Computation that is the title of this lecture note is a research field mathematically modeling physics in natural science, events in social science, and computing these mathematical models. This field has been studied by applied mathematicians, scientists, and engineers. Originally Mathematical Science is the research field for studying mathematical models. We know some subdivided fields in the Mathematical Science such as Mathematical Physics, Mathematical Biology, Mathematical Sociology, Mathematical Economics, Mathematical Psychology, and so on. Especially, the Mathematical Biology is a typical research field proposing well-known mathematical models. Hodgkin-Huxley model [1] is a famous model of physiology which can simulates a pulsative signal of ion going in and out nerve membrane. Hodgkin and Huxley got the Nobel Prize of Physiology and Medicine in 1963 by the achievement of this study. This model was further extended by FitzHugh and Nagumo as FitzHugh-Nagumo model [2]. Kermack-McKendrick model [3]-[5] is known as the model of Immunology simulating infectious diseases. This is also called SIR model named from the first letter of three words: Susceptible, Infectious and Recovered. Lotka-Volterra model [6] is a model of Ecology simulating a competition system such as pray-predator system among fungi, plant, insect, animal, and human. This model is also applied to problems in resource-consumer system. Lotka-Volterra model can be reduced to a system of logistic equations simulating population movements. Also in Economy, Black-Scholes equation has been employed to predict the stock market quotations. Currently traffic jam, internet traffic, fashion, rumor, ⋯ are being predicted by their own models or Artificial Intelligence (AI) using their big data. As more famous mathematical models, Maxwell's equations, Schroedinger's equation, and Navier-Stokes equations are the most complete and complex models for electro magnetics, quantum mechanics, and fluid mechanics in physics.

In this lecture note, the basic of mathematical modeling and the computation is introduced. The mathematical model is based on a partial or ordinary differential equation, or the set of equations. The models are computed by a finite-difference method. A macro program using an Excel sheet for simulating 2D problems and Fortran programs simulating 3D problems which can be downloaded from the Web page are introduced to assist the lecture.

2. Entrance of Mathematical Modeling and Computation

2.1 Reaction Equation

Differentiating e^{-t} results in $-e^{-t}$. If $u = e^{-t}$, then $u' = -e^{-t}$ and $u' = -u$ ($du/dt = -u$). This equation is called 'Ordinary Differential Equation (ODE)'. The differential term du/dt is called 'Derivative function' of unknown variable u with respect to time t.

Finite Difference Method (FDM) is a numerical method which is widely employed as the basis of the mathematical modeling and computation.

Now let's solve $du/dt = -u$ using FDM. Then we can make the following approximated equation:

$$\frac{\left(u^{n+1} - u^{n}\right)}{\Delta t} = -u^{n} \tag{2.1}$$

where Δt is the time interval and n is the time-step (not exponentiation) in time direction (the detail is explained in later). The unknown variable u at n time-step is defined as u^n.

Reforming Eq. (2.1), we obtain

$$u^{n+1} = u^{n} - \Delta t\, u^{n} \tag{2.2}$$

This equation is equivalent to an assignment statement in computer programs for updating u^n to u^{n+1} by increasing n as 1,2,3,・・・. For example, if $t = 0$ and $u = 1$ at $n = 0$; $\Delta t = 0.1$, then we can obtain the following results for u.

n	t	u	e^{-t}
0	0.0	1.0000	1.0000
1	0.1	0.9000	0.9048
2	0.2	0.8100	0.8187
3	0.3	0.7290	0.7408
4	0.4	0.6561	0.6703
5	0.5	0.5905	0.6065
6	0.6	0.5314	0.5488
7	0.7	0.4783	0.4966
8	0.8	0.4305	0.4493
9	0.9	0.3874	0.4066
10	1.0	0.3487	0.3679
11	1.1	0.3138	0.3329
12	1.2	0.2824	0.3012
13	1.3	0.2542	0.2725
14	1.4	0.2288	0.2466
15	1.5	0.2059	0.2231
16	1.6	0.1853	0.2019
17	1.7	0.1668	0.1827
18	1.8	0.1501	0.1653
19	1.9	0.1351	0.1496
20	2.0	0.1216	0.1353

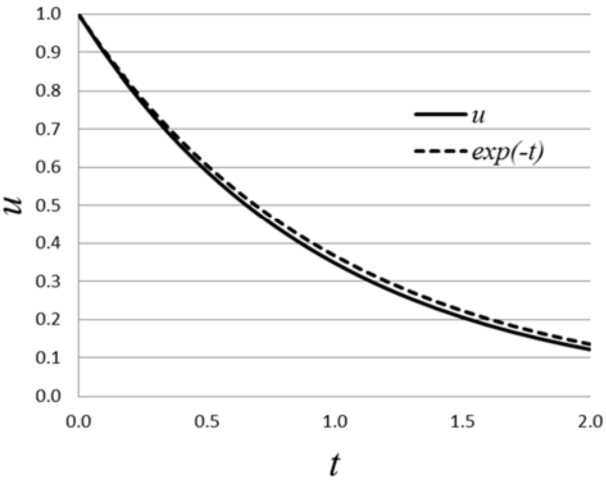


Fig.2.1 Calculated results of u compared with the exact solution e^{-t}.

While increasing in the number of n, u asymptotically approaches zero. The right figure in Fig. 2.1 indicates that the value of u is almost the same with e^{-t}, because an exact solution of u is e^{-t}. FDM solution can approximate the value.

$du/dt = -u$ as ODE is alternatively called 'Reaction Equation'. Reaction means the chemical reaction. Assuming u as a chemical species, u approaching zero indicates that the species u changes to another species. The left- and right-hand sides of reaction equations are respectively called 'Time-derivative Term' and 'Reaction Term'.

On the other hand, if $du/dt = u$, then an exact solution is e^{t}. While increasing in the number of n, u increases exponentially. Such a chemical reaction or an infectious disease may be unrealistic.

2-2 System of Reaction Equations

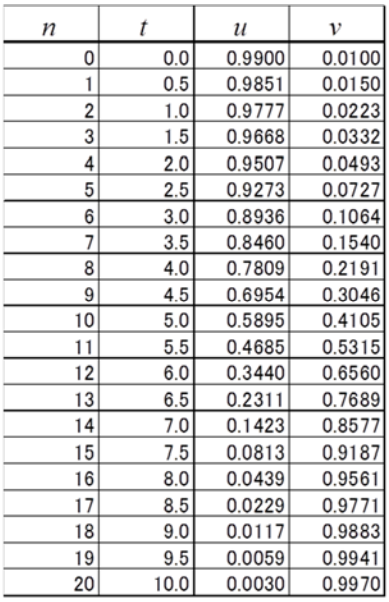
Let's solve two reaction equations simultaneously to simulate a chemical reaction between two species. If the densities are defined by u and v, we can derive the following system of reaction equations:

$$\begin{aligned} \frac{du}{dt} &= -vu \\ \frac{dv}{dt} &= uv \end{aligned} \tag{2.3}$$

Using FDM,

$$\begin{aligned} u^{n+1} &= u^{n} - \Delta t\, v^{n} u^{n} \\ v^{n+1} &= v^{n} + \Delta t\, u^{n} v^{n} \end{aligned} \tag{2.4}$$

If $u = 0.99$ and $v = 0.01$ at $t = 0$ as initial conditions; $\Delta t = 0.5$, then the FDM solutions are obtained as



n	*t*	*u*	*v*
0	0.0	0.9900	0.0100
1	0.5	0.9851	0.0150
2	1.0	0.9777	0.0223
3	1.5	0.9668	0.0332
4	2.0	0.9507	0.0493
5	2.5	0.9273	0.0727
6	3.0	0.8936	0.1064
7	3.5	0.8460	0.1540
8	4.0	0.7809	0.2191
9	4.5	0.6954	0.3046
10	5.0	0.5895	0.4105
11	5.5	0.4685	0.5315
12	6.0	0.3440	0.6560
13	6.5	0.2311	0.7689
14	7.0	0.1423	0.8577
15	7.5	0.0813	0.9187
16	8.0	0.0439	0.9561
17	8.5	0.0229	0.9771
18	9.0	0.0117	0.9883
19	9.5	0.0059	0.9941
20	10.0	0.0030	0.9970

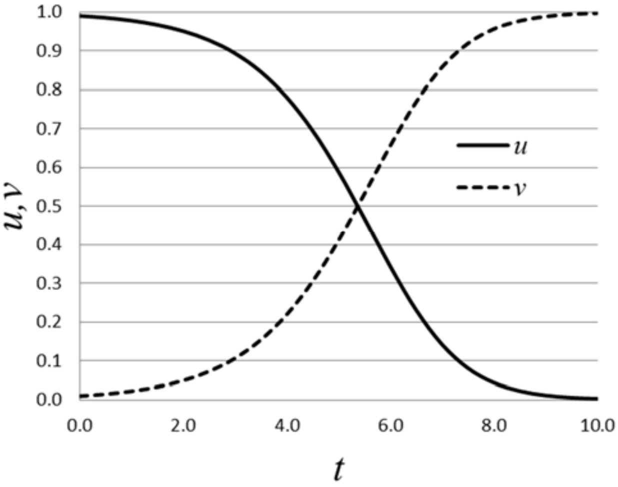


Fig. 2.2 Time-dependent solutions for u and v.

When n = 18, the values of u and v are almost reversed from the initial values. These values further asymptotically reach u = 0 and v = 1 while increasing in the number of n. Such a relationship may be also observed in a lot of physics and social events where some things change to some others.

As a well-known mathematical model based on three reaction equations, a transmission model of infection, is briefly introduced here. This model was proposed by Kermack and McKendrick [1] - [3] as the following three equations:

$$\begin{aligned}\frac{dS(t)}{dt} &= -\beta\, S(t)I(t) \\ \frac{dI(t)}{dt} &= -\gamma\, I(t) + \beta\, S(t)I(t) \\ \frac{dR(t)}{dt} &= \gamma\, I(t)\end{aligned} \tag{2.5}$$

where $S(t)$ is the density of population under susceptible condition. $I(t)$ is the density of population under infectious condition. $R(t)$ is the density of population under recovered, removed or immune condition. β is the transmission rate from susceptible condition to infectious condition and γ is the recovered rate; both are empirical constants. The total population $N(t) = S(t) + I(t) + R(t)$ is constant and the above equations derives

$$\frac{dS(t)}{dt} + \frac{dI(t)}{dt} + \frac{dR(t)}{dt} = 0 \tag{2.6}$$

This model is alternatively called ‘SIR Model’ using the letter of parameters: $S(t)$, $I(t)$ and $R(t)$. Now setting $\Delta t = 1.0$, $\beta = 0.4$, $\gamma = 0.03$, $S(0) = 999$, $I(0) = 1$ and $R(0) = 0$ as initial conditions, we obtain the following results:

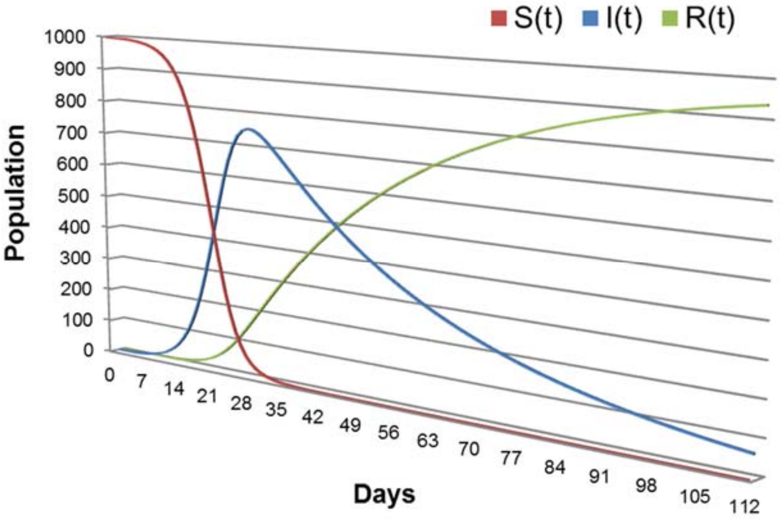


Fig.2.3 Time-dependent solutions for $S(t)$, $I(t)$ and $R(t)$.

This figure indicates that the population under infectious condition increases rapidly within first four weeks from only one person under infectious condition. The number turns to decrease after the four weeks and the population under recovered condition increases according to the decrease of population under infectious condition. We may experience a similar trend when influenza is prevalent. SIR model has been further modified to more practical models while considering birth and death rates, vaccine effect, reinfection, latent period and so on (shown in later at Chapter 4).

2-3 Partial Differential Equation

The independent variable t in the reaction equation $du/dt = -u$ was the time. ODE has one independent variable, while Partial Differential Equation (PDE) is the differential equation that has more than two independent variables.

For example, the following equation is the simplest PDE:

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x} \tag{2.7}$$

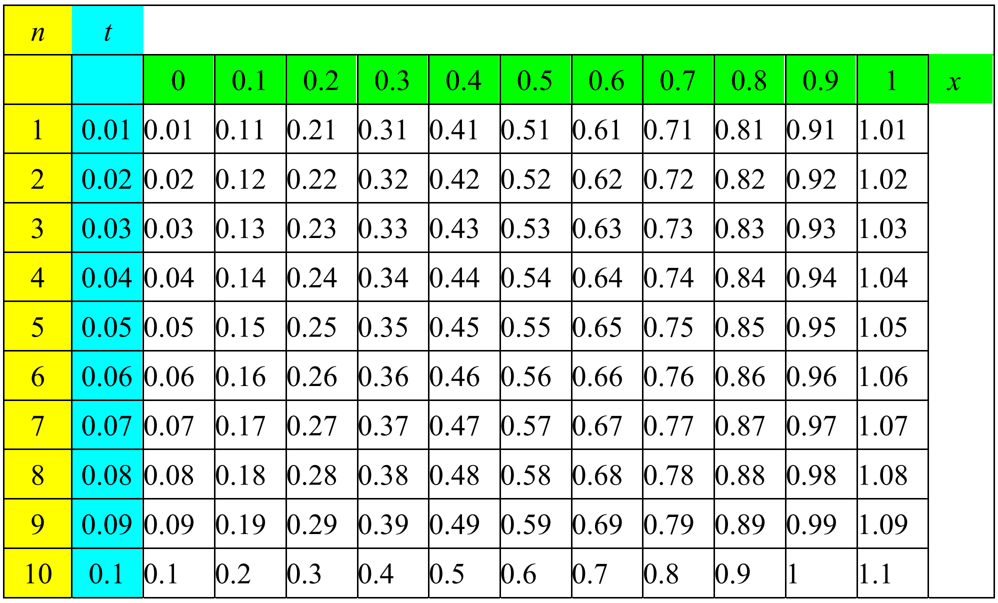
This PDE is called ‘Convection Equation’. We use the symbol ’∂’ for the derivatives in PDE instead of ’d’ in ODE. The differential terms $\partial u/\partial t$ and $\partial u/\partial x$ are called ‘Partial Derivative’. We can easily find one of the solutions for Eq. (2.7) as $u = x + t$. Then $\partial u/\partial t$ = $\partial u/\partial x$ =1. Let’s solve this problem using FDM. The convection equation is approximated by FDM as

$$\frac{u^{n+1}(x)-u^{n}(x)}{\Delta t}=\frac{u^{n}(x+\Delta x)-u^{n}(x-\Delta x)}{2\Delta x} \tag{2.8}$$

where x is one-dimensional space coordinate, u (x) is the unknown variable at x and Δx is the space interval between two neighboring grid points. In this equation, $\partial u/\partial x$ is approximated by the second-order central difference. $\partial u/\partial t$ is approximated by the first-order explicit Euler forward difference. These approximations are based on the Taylor's expansion. The detail is explained in later (see Section 2.7). The method in which unknown variables are calculated only from known values is called 'Explicit Method.' $u^{n+1}(x)$ is calculated by the following assignment expression deformed from Eq. (2.8):

$$u^{n+1}(x)=u^{n}(x)+\frac{\Delta t}{2\Delta x}\left[u^{n}(x+\Delta x)-u^{n}(x-\Delta x)\right] \tag{2.9}$$

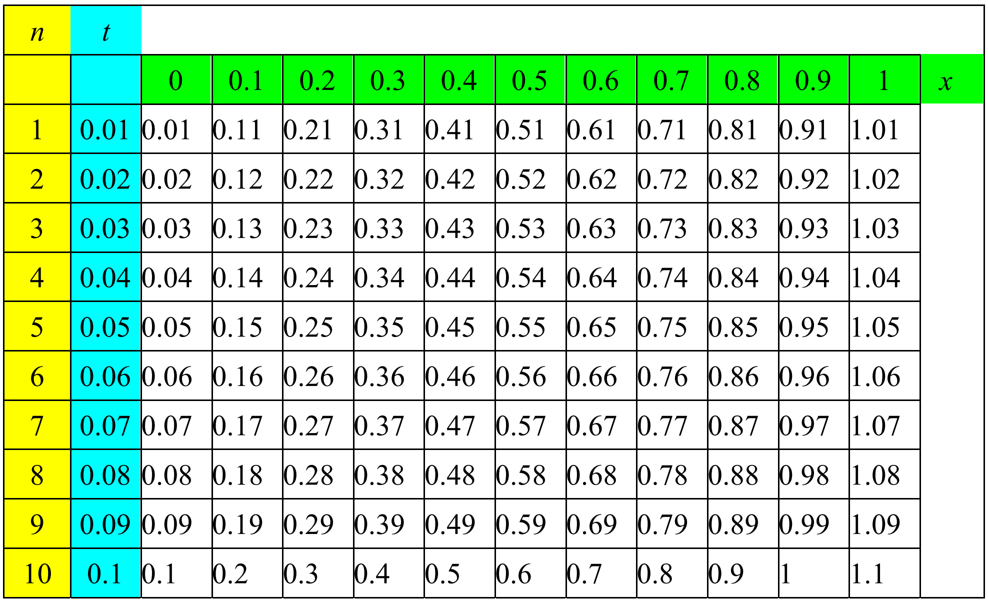
Now let's calculate Eq. (2.9). We set $\Delta x = 0.1$ at $0 \leqq x \leqq 1$; the exact solution $u = x + t$ at $x = 0$ and $x = 1$ as the boundary conditions. The exact solution $u = x + t$ is specified at all points from $x = 0$ to $x = 1$ at $t = 0$ as the initial conditions. Then the solutions are obtained as in the following sheet according to the increase in n by $\Delta t = 0.01$:



n	t												
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	x
1	0.01	0.01	0.11	0.21	0.31	0.41	0.51	0.61	0.71	0.81	0.91	1.01	
2	0.02	0.02	0.12	0.22	0.32	0.42	0.52	0.62	0.72	0.82	0.92	1.02	
3	0.03	0.03	0.13	0.23	0.33	0.43	0.53	0.63	0.73	0.83	0.93	1.03	
4	0.04	0.04	0.14	0.24	0.34	0.44	0.54	0.64	0.74	0.84	0.94	1.04	
5	0.05	0.05	0.15	0.25	0.35	0.45	0.55	0.65	0.75	0.85	0.95	1.05	
6	0.06	0.06	0.16	0.26	0.36	0.46	0.56	0.66	0.76	0.86	0.96	1.06	
7	0.07	0.07	0.17	0.27	0.37	0.47	0.57	0.67	0.77	0.87	0.97	1.07	
8	0.08	0.08	0.18	0.28	0.38	0.48	0.58	0.68	0.78	0.88	0.98	1.08	
9	0.09	0.09	0.19	0.29	0.39	0.49	0.59	0.69	0.79	0.89	0.99	1.09	
10	0.1	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	

Fig. 2.4(a) Solutions by FDM (Δt = 0.01)

The exact solutions are also obtained as



n	*t*												
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	*x*
1	0.01	0.01	0.11	0.21	0.31	0.41	0.51	0.61	0.71	0.81	0.91	1.01	
2	0.02	0.02	0.12	0.22	0.32	0.42	0.52	0.62	0.72	0.82	0.92	1.02	
3	0.03	0.03	0.13	0.23	0.33	0.43	0.53	0.63	0.73	0.83	0.93	1.03	
4	0.04	0.04	0.14	0.24	0.34	0.44	0.54	0.64	0.74	0.84	0.94	1.04	
5	0.05	0.05	0.15	0.25	0.35	0.45	0.55	0.65	0.75	0.85	0.95	1.05	
6	0.06	0.06	0.16	0.26	0.36	0.46	0.56	0.66	0.76	0.86	0.96	1.06	
7	0.07	0.07	0.17	0.27	0.37	0.47	0.57	0.67	0.77	0.87	0.97	1.07	
8	0.08	0.08	0.18	0.28	0.38	0.48	0.58	0.68	0.78	0.88	0.98	1.08	
9	0.09	0.09	0.19	0.29	0.39	0.49	0.59	0.69	0.79	0.89	0.99	1.09	
10	0.1	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	

Fig. 2.4(b) Exact solutions

2-4 Diffusion Equation

A number of physics and social events may be governed by diffusion phenomena. Not only heat conduction but also smell, fashion, rumor, infection, vegetation and so on, are dominated by a kind of diffusions. PDE governing a diffusion is called 'Diffusion Equation.' One of typical diffusion equations is defined by

$$\frac{\partial u}{\partial t}=\frac{\partial^2 u}{\partial x^2} \tag{2.10}$$

The right-hand side has a second partial derivative of u with respect to x. This equation can simulate a time-dependent diffusion. For example, one-dimensional heat conduction is governed by the following similar equation:

$$\frac{\partial u}{\partial t}=\kappa\frac{\partial^2 u}{\partial x^2} \tag{2.11}$$

where κ is the heat-conductivity coefficient. This equation is approximated by FDM as

$$\frac{u^{n+1}(x)-u^n(x)}{\Delta t}=\kappa\frac{u^n(x+\Delta x)-2u^n(x)+u^n(x-\Delta x)}{(\Delta x)^2} \tag{2.12}$$

The right-hand side is derived by the second-order central difference (see Section 2.7). $u^{n+1}(x)$ is calculated by the following assignment expression:

$$u^{n+1}(x)=u^n(x)+\frac{\kappa\Delta t}{(\Delta x)^2}\left[u^n(x+\Delta x)-2u^n(x)+u^n(x-\Delta x)\right] \tag{2.13}$$

Now let's consider a 1-D heat conductive problem setting 10℃ as the initial condition at $0 \leqq x \leqq 10$. As boundary conditions, the left edge at $x = 0$ is fixed to 10℃ and the right edge at $x = 10$ is heated and fixed to 100℃. Then the heat is gradually diffused from $x = 10$ toward $x = 0$. If $\Delta x = 1$, $\Delta t = 0.5$ and $\kappa = 1$, we obtain the solutions till $n = 10$ as shown in Fig. 2.5(a):

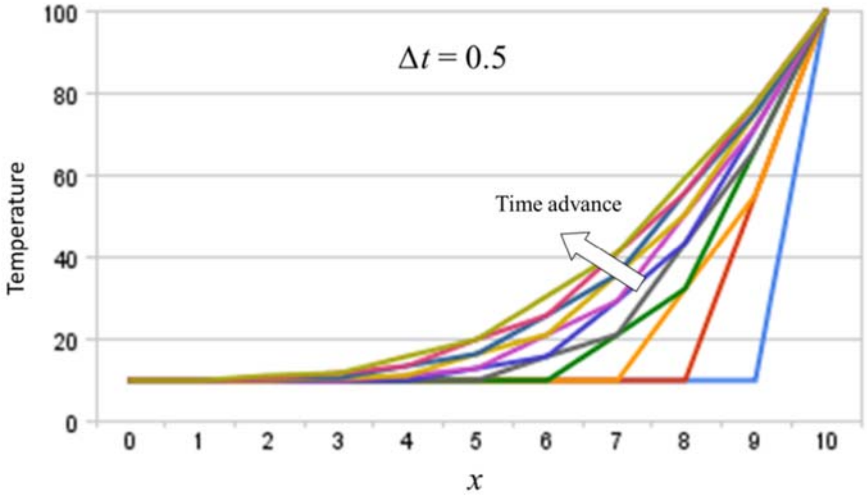


Fig. 2.5(a) Time-dependent solutions (Δt = 0.5).

Temperature increases gradually from $x = 0$ toward $x = 10$ while increasing in time. The same calculation in which only Δt is slightly increased to 0.6 obtains the following results as in Fig. 2.5(b):

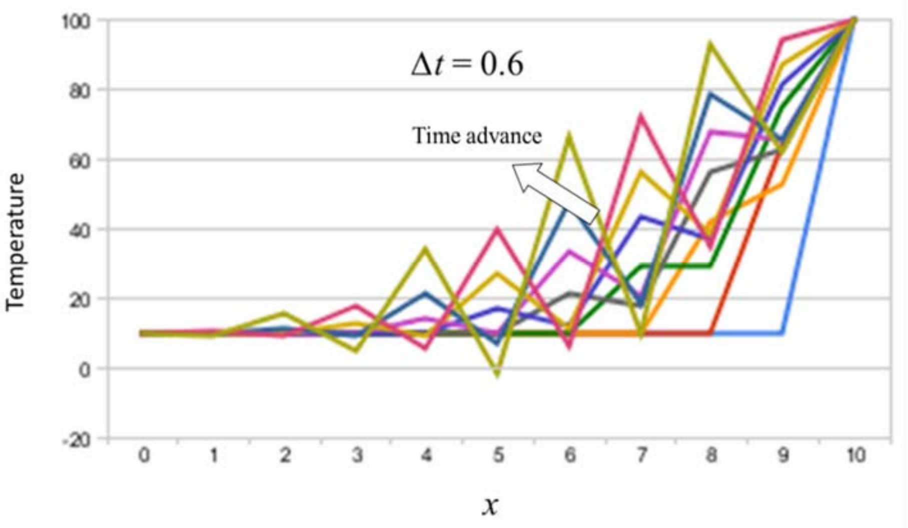


Fig. 2.5(b) Time-dependent solutions (Δt = 0.6).

The obtained temperature oscillates unphysically at every time step. The oscillation does not demonstrate any real physics, but a numerical error. Such oscillation is called 'Numerical

Oscillation.' The numerical error can be explained by the linear stability analysis. $\Delta t = 0.5$ could get a stable solution and that using $\Delta t = 0.6$ became unstable. A numerical method satisfying the linear stability (see Section 3.3) must be employed to get a stable solution.

2-5 Types of PDE

Let's define a quasi-linear second PDE for unknown variable $u(x, y)$ with respect to two independent variables x, y in a general form:

$$A\frac{\partial^2 u}{\partial x^2} + 2B\frac{\partial^2 u}{\partial x \partial y} + C\frac{\partial^2 u}{\partial y^2} = f\left(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right) \tag{2.14}$$

where the coefficients A, B and C are constants, and f at right-hand side is at most a function of x, y and first partial derivatives.

Defining $D = B^2 - AC$, this equation is categorized to three types according to the sign of D:

Hyperbolic	$(D > 0)$
Parabolic	$(D = 0)$
Elliptic	$(D < 0)$

Originally D is the element in the root square derived in the roots for the following characteristic equation of Eq. (2.14) (see Section 3.9) :

$$A\left(\frac{dy}{dx}\right)^2 - 2B\left(\frac{dy}{dx}\right) + C = 0 \tag{2.15}$$

where the roots of Eq. (2.15) are obtained as

$$\frac{dy}{dx} = \frac{B \pm \sqrt{B^2 - AC}}{A} \tag{2.16}$$

Eq. (2.16) indicates that PDE is classified into hyperbolic, parabolic and elliptic types if the roots are two different real values, duplicate value, and two different conjugate complex values, respectively.

For example, each type of PDE has a typical form:

Hypersonic: $$\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = 0$$

Parabolic: $$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial y} \tag{2.17}$$

Elliptic: $$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0$$

The hyperbolic form is identical to PDE called 'Wave Equation'. The form for parabolic PDE becomes the same with the equation of heat conduction if y is changed to the time t. The elliptic PDE is called 'Laplace Equation' and that with a value at right-hand side is called 'Poisson Equation'.

2-6 Finite Difference of Laplace Equation

2-D Laplace equation is defined by

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0 \tag{2.18}$$

This equation is approximated by the second-order central difference as follows:

$$\frac{u(x+\Delta x,y)-2u(x,y)+u(x-\Delta x,y)}{(\Delta x)^2}+\frac{u(x,y+\Delta y)-2u(x,y)+u(x,y-\Delta y)}{(\Delta y)^2}=0 \tag{2.19}$$

where $u(x,y)$ is the unknown variable u at the point (x,y). Δx and Δy are space intervals between two neighboring grid points in x and y directions. Setting $\Delta x = \Delta y$, we further simplify Eq. (2.19) into

$$u(x,y)=\frac{1}{4}\left[u(x+\Delta x,y)+u(x-\Delta x,y)+u(x,y+\Delta y)+u(x,y-\Delta y)\right] \tag{2.20}$$

This equation has quite a simple form for updating $u(x,y)$ as the sum of four known values around the point for $u(x,y)$.

2-7 Taylor's Expansion

We know that the first derivative term *du/dx* of *u* with respect to *x* is originally defined by

$$\frac{du}{dx}=\lim_{\Delta x\to 0}\frac{u(x+\Delta x)-u(x)}{\Delta x} \tag{2.21}$$

where $u(x)$ and $u(x+\Delta x)$ are the unknown variables on a grid point shown in the following figure:

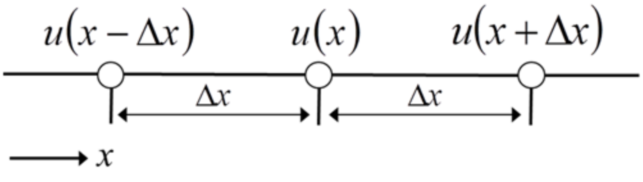


Fig.2.6 Definition of unknown variables

Finite-difference approximation takes the truncation error into account for *du/dx*. $u(x+\Delta x)$ is derived by Taylor's expansion as

$$u(x+\Delta x)=u(x)+\Delta x\frac{du(x)}{dx}+\frac{(\Delta x)^2}{2!}\frac{d^2u(x)}{dx^2}+\Lambda+\frac{(\Delta x)^{n-1}}{(n-1)!}\frac{d^{n-1}u(x)}{dx^{n-1}}+\frac{(\Delta x)^n}{n!}\frac{d^nu(\xi)}{dx^n}$$

(2.22)

where $x<\xi<x+\Delta x$. We can obtain *du/dx* from Eq. (2.22):

$$\frac{du(x)}{dx}=\frac{u(x+\Delta x)-u(x)}{\Delta x}-\frac{\Delta x}{2!}\frac{d^2u(x)}{dx^2}-\Lambda-\frac{(\Delta x)^{n-2}}{(n-1)!}\frac{d^{n-1}u(x)}{dx^{n-1}}-\frac{(\Delta x)^{n-1}}{n!}\frac{d^nu(\xi)}{dx^n}$$

(2.23)

Terms after the second term at the right-hand side correspond to the truncation error. These terms have a first-order error because of $\mathrm{O}(\Delta x)$. Then this equation is rewritten by

$$\frac{du(x)}{dx}=\frac{u(x+\Delta x)-u(x)}{\Delta x}+\mathrm{O}(\Delta x) \tag{2.24}$$

Also $u(x-\Delta x)$ can be derived by Taylor's expansion as the same manner with Eq. (2.22) as

$$u(x-\Delta x)=u(x)+(-\Delta x)\frac{du(x)}{dx}+\frac{(-\Delta x)^2}{2!}\frac{d^2u(x)}{dx^2}+\Lambda+\frac{(-\Delta x)^{n-1}}{(n-1)!}\frac{d^{n-1}u(x)}{dx^{n-1}}+\frac{(-\Delta x)^n}{n!}\frac{d^nu(\xi)}{dx^n}$$

(1.25)

We can obtain an alternative form with the second-order truncation error using $u(x+\Delta x)$ and $u(x-\Delta x)$:

$$\begin{aligned}\frac{du(x)}{dx}&=\frac{u(x+\Delta x)-u(x-\Delta x)}{2\Delta x}-\frac{(\Delta x)^2}{12}\frac{d^3u}{dx^3}+\Lambda\\&=\frac{u(x+\Delta x)-u(x-\Delta x)}{2\Delta x}+\mathrm{O}\left[(\Delta x)^2\right]\end{aligned} \tag{2.26}$$

Now let's modify the notation of unknown variables and the grid points where the unknown values are located using the number of grid points j as shown in the following figure:

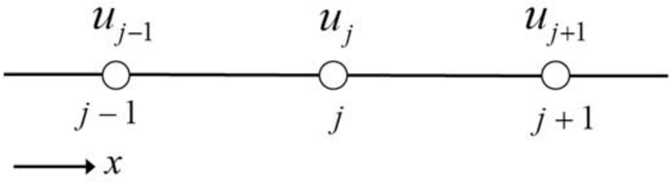


Fig. 2.7 Definition of grid points and unknown variables

Removing the truncation error term $O(\varDelta x)$, the first-order forward and backward differences of *du/dx* are defined by

$$\left(\frac{du}{dx}\right)_j = \frac{u_{j+1} - u_j}{\Delta x}, \quad \left(\frac{du}{dx}\right)_j = \frac{u_j - u_{j-1}}{\Delta x} \tag{2.27}$$

Also the second-order central difference is defined by

$$\left(\frac{du}{dx}\right)_j = \frac{u_{j+1} - u_{j-1}}{2\Delta x} \tag{2.28}$$

The second-order central difference of d^2u/dx^2 can be further derived as

$$\left(\frac{d^2u}{dx^2}\right)_j = \frac{u_{j+1} - 2u_j + u_{j-1}}{(\Delta x)^2} \tag{2.29}$$

Note that the second derivative term in Eq. (2.29) is originally defined by

$$\left(\frac{d^2u}{dx^2}\right)_j = \left[\frac{d}{dx}\left(\frac{du}{dx}\right)\right]_j \tag{2.30}$$

The second derivative term is firstly approximated using the first derivatives at the intermediate points $j \pm 1/2$ as shown in the following figure:

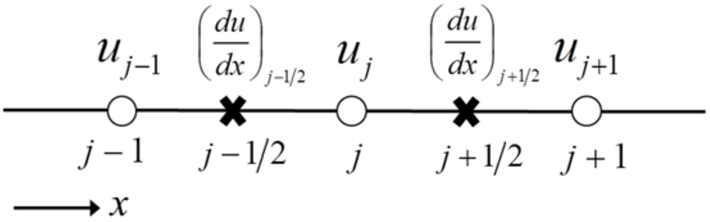

u_{j-1}
$\left(\frac{du}{dx}\right)_{j-1/2}$
u_j
$\left(\frac{du}{dx}\right)_{j+1/2}$
u_{j+1}
$j-1$
$j-1/2$
j
$j+1/2$
$j+1$
x

Fig. 2.8 Definition of grid points and first derivatives at $j \pm 1/2$.

Using the first derivatives, we obtain

$$\left(\frac{d^2u}{dx^2}\right)_j = \frac{\left(\frac{du}{dx}\right)_{j+1/2} - \left(\frac{du}{dx}\right)_{j-1/2}}{\Delta x} \tag{2.31}$$

where the first derivatives are approximated by the second-order central difference as

$$\left(\frac{du}{dx}\right)_{j+1/2}=\frac{u_{j+1}-u_j}{\Delta x} \quad \text{and} \quad \left(\frac{du}{dx}\right)_{j-1/2}=\frac{u_j-u_{j-1}}{\Delta x} \tag{2.32}$$

Finally we can obtain

$$\left(\frac{d^2u}{dx^2}\right)_j=\frac{u_{j+1}-2u_j+u_{j-1}}{(\Delta x)^2} \tag{2.33}$$

2-8 Boundary Condition

Because we calculate unknown variables on grid points at a finite area for solving PDE by FDM, we should specify the Boundary Conditions (B.C.). The condition setting unknown variable itself at the boundary is called 'Dirichlet's B.C.', while the boundary condition specifying a first partial derivative of unknown variable is called 'Neumann's B.C.'. Since the Neumann's B.C. specifies the derivative of unknown variables, we need to get the solution of unknown variable at the boundary, for example, at the grid point $j=1$ in Fig. 2.9. Let's specify $(\partial u/\partial x)_1 = 0$ at $j=1$. One of typical treatment for the B.C. is first to derive a first-order forward difference of $(\partial^2 u/\partial x^2)_1$ according to Fig. 2.9. Then we derive the FDM form:

$$\left(\frac{d^2u}{dx^2}\right)_1=\frac{\left(\frac{du}{dx}\right)_{1+1/2}-\left(\frac{du}{dx}\right)_1}{\Delta x/2} \tag{2.34}$$

The first derivatives are further approximated by

$$\left(\frac{du}{dx}\right)_{1+1/2}=\frac{u_2-u_1}{\Delta x} \quad \text{and} \quad \left(\frac{du}{dx}\right)_1=0 \tag{2.35}$$

Substituting Eq. (2.35) into Eq. (2.34), we obtain

$$\left(\frac{d^2u}{dx^2}\right)_1=\frac{2(u_2-u_1)}{(\Delta x)^2} \tag{2.36}$$

The derivative at the right edge boundary can be derived as the same manner.

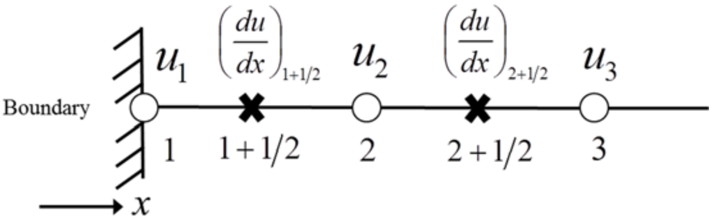

Boundary
u_1
$\left(\frac{du}{dx}\right)_{1+1/2}$
u_2
$\left(\frac{du}{dx}\right)_{2+1/2}$
u_3
1
1+1/2
2
2+1/2
3
x

Fig. 2.9 Definition of grid points near the left boundary.

Note that zeroth-order accurate B.C. setting $u_1 = u_2$ may be also acceptable and rather usual as a simpler treatment for Neumann's B.C..

3. FDM for Mathematical Models based on PDE

Let's explain the detail of FDM applied to mathematical models defined by PDE hereafter.

3-1 Relaxation Method for Laplace Equation

Following figure shows a computational grid in two dimensions. FDM solves equations only on the grid points. If we number *i* (1 to *IF*) and *j* (1 to *JF*) in *x* and y directions on the grid points, then $u_{i,j}$ is defined as the unknown variable *u* on (*i* , *j*) grid point.

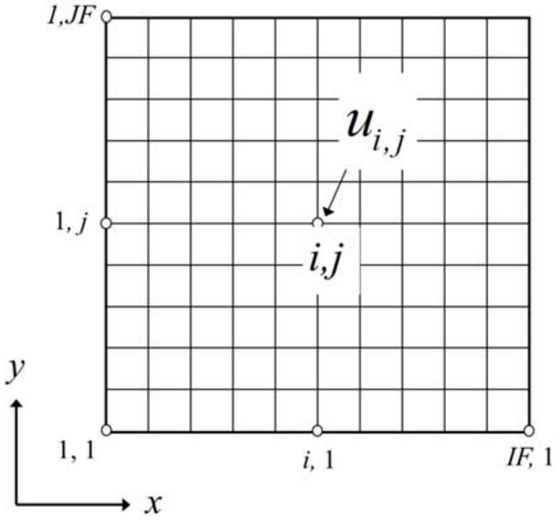


Fig. 3.1 Definition of 2-D grid points and unknown variables

2-D Laplace equation Eq. (2.18) is approximated by the second-order central difference on the grid points as

$$\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{(\Delta x)^2}+\frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{(\Delta y)^2}=0 \tag{3.1}$$

In particular, $\Delta x = \Delta y$ further simplifies the equation into

$$u_{i,j}=\frac{1}{4}\left(u_{i+1,j}+u_{i-1,j}+u_{i,j+1}+u_{i,j-1}\right) \tag{3.2}$$

Using the iteration number *n* (note that *n* is not the time step here because of no time derivative in Laplace equation), we obtain a well-known relaxation method called 'Jacobi method':

$$u_{i,j}^{n+1}=\frac{1}{4}\left(u_{i+1,j}^{n}+u_{i-1,j}^{n}+u_{i,j+1}^{n}+u_{i,j-1}^{n}\right) \tag{3.3}$$

A steady state solution is only obtained by increasing the iteration number *n* and repeating the computation in Eq. (3.3). Jacobi method calculates unknown variable u^{n+1} at (*i,j*) grid point from already known values u^n at four grid points around the grid point (i,j) . This means that all unknown variables u^{n+1} can be calculated in parallel from only known values. Jacobi method is straightforward to a parallel computation for Eq. (3.1) if we use a parallel computer.

The following is a further modification on Jacobi method at the point for saving the computational time assuming that we use a computer with a single processor or at most a few processors. Gauss-Seidel (GS) method is a relaxation method modified from Jacobi method to reduce the iteration number n, accelerating the computational time. GS method is defined by

$$u_{i,j}^{n+1} = \frac{1}{4}\left(u_{i+1,j}^{n} + u_{i-1,j}^{n+1} + u_{i,j+1}^{n} + u_{i,j-1}^{n+1}\right) \tag{3.4}$$

Compared with Jacobi method, two terms at the right-hand side are defined at the iteration number n+1.

Figure 3.2 shows the grid points and a hyper line (skew line crossing the grid points) which explains how to calculate Eq. (3.4). The hyper line moves from the lower left grid points toward the upper right grid points. The unknown variables at grid points where the hyper line passed are updated to those at n+1. GS method applies the already updated variables $u_{i-1,j}^{n+1}$ and $u_{i,j-1}^{n+1}$ into the equation as in Eq. (3.4) for calculating $u_{i,j}^{n+1}$. Using these variables in GS method resultantly can reduce the iteration number to half as compared with that of Jacobi method.

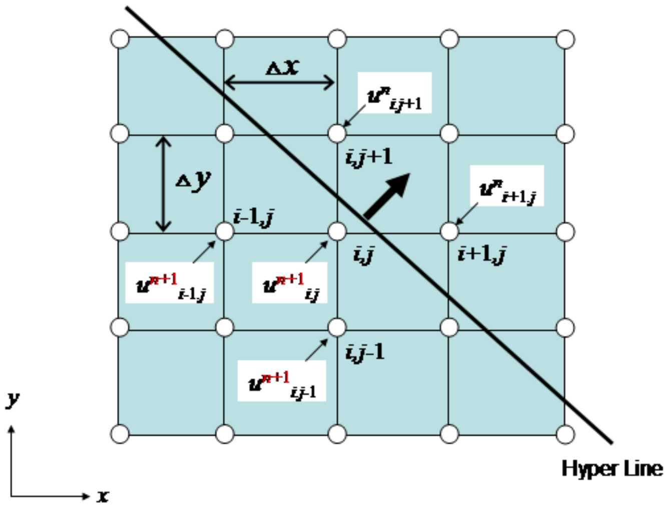


Fig. 3.2 Schematic of GS method and hyper line.

Successive Over-Relaxation (SOR) method [7] is a relaxation method further extended from GS method. Now if $u_{i,j}^{n+1}$ in Fig. (3.4) is rewritten as $\left(u_{i,j}^{n+1}\right)_{GS}$, SOR method is defined by

$$u_{i,j}^{n+1} = \omega\left(u_{i,j}^{n+1}\right)_{GS} + \left(1-\omega\right)u_{i,j}^{n} \tag{3.5}$$

This equation represents the linear interpolation between $u_{i,j}^{n}$ and $\left(u_{i,j}^{n+1}\right)_{GS}$. We know that the value of interpolation parameter ω is generally defined within $0 < \omega < 1$. But this value is set to more than one ($1 < \omega$) and $\omega < 2$ in SOR method. This value is called 'Over-relaxation Parameter'. It means that the value of GS method is overestimated in SOR method. Empirically ω is usually set to around 1.5. If $\omega = 1.5$, SOR method can reduce the iteration number to one third from that of Jacobi method. SOR method may be the most proper method for solving Laplace equation.

3-2 3-D Potential Flow solved by SOR Method

Let's solve a 3-D potential flow using SOR method. Flow velocities u, v, w in x, y and z directions is assumed to be defined by the gradient of potential ϕ for the potential flow as follows:

$$u = \frac{\partial \phi}{\partial x},\quad v = \frac{\partial \phi}{\partial y},\quad w = \frac{\partial \phi}{\partial z} \tag{3.6}$$

The continuity equation governing 3-D incompressible flows is defined by

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{3.7}$$

Substituting Eq. (3.6) into Eq. (3.7), then we obtain

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2} = 0 \tag{3.8}$$

and Eq. (3.8) results in 3-D Laplace equation.

Vorticities of 3-D incompressible flows are defined by the following vector:

$$\left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z},\quad \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x},\quad \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right)$$

Since all these elements are reduced to zero if Eq. (3.6) is substituted, the potential flow is equivalent to an irrotational flow.

The second-order finite-difference form of Eq. (3.8) is derived as

$$\frac{\phi_{i+1,j,k}-2\phi_{i,j,k}+\phi_{i-1,j,k}}{(\Delta x)^2}+\frac{\phi_{i,j+1,k}-2\phi_{i,j,k}+\phi_{i,j-1,k}}{(\Delta y)^2}+\frac{\phi_{i,j,k+1}-2\phi_{i,j,k}+\phi_{i,j,k-1}}{(\Delta z)^2}=0 \quad (3.9)$$

where the subscript k is the number of grid points in z direction. Eq. (3.9) is further rewritten as the following equation:

$$\phi_{i,j,k}=\frac{1}{L}\left[\frac{\phi_{i+1,j,k}+\phi_{i-1,j,k}}{(\Delta x)^2}+\frac{\phi_{i,j+1,k}+\phi_{i,j-1,k}}{(\Delta y)^2}+\frac{\phi_{i,j,k+1}+\phi_{i,j,k-1}}{(\Delta z)^2}\right] \quad (3.10)$$

where $L=2/(\Delta x)^2+2/(\Delta y)^2+2/(\Delta z)^2$. If SOR method is applied to Eq. (3.10), then we obtain the following final form:

$$\phi^{n+1}_{i,j,k}=(1-\omega)\phi^{n}_{i,j,k}+\frac{\omega}{L}\left[\frac{\phi^{n}_{i+1,j,k}+\phi^{n+1}_{i-1,j,k}}{(\Delta x)^2}+\frac{\phi^{n}_{i,j+1,k}+\phi^{n+1}_{i,j-1,k}}{(\Delta y)^2}+\frac{\phi^{n}_{i,j,k+1}+\phi^{n+1}_{i,j,k-1}}{(\Delta z)^2}\right] \quad (3.11)$$

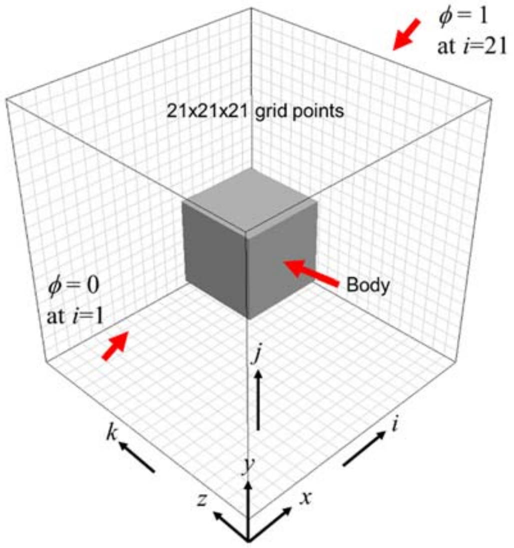
The solution of 3-D potential flow is obtained by the iterative computation.

The problem solved here is shown in Fig. 3.3(a). The computational grid points are $21\times21\times21$. A potential flow over a cubic which surfaces are located at $i=j=k=8$ and $i=j=k=14$ is considered. As the boundary conditions, $\phi=0$ at the inlet boundary surface $i=1$, $\phi=1$ at the outlet boundary surface $i=21$. $\partial\phi/\partial n=0$ is specified at the normal direction of the other surfaces. $\Delta x=\Delta y=\Delta z=0.1$. ω is fixed to 1.5.

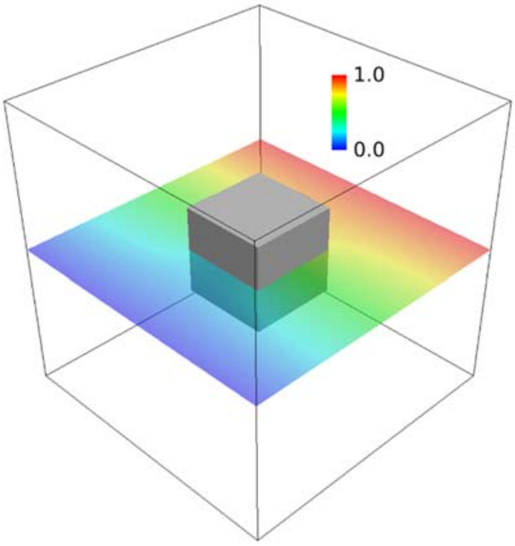
Fig. 3.3(b) shows the calculated potential distribution on $j=11$ grid surface. The potential is smoothly distributed from $\phi=0$ to $\phi=1$ over the cubic.

Fig. 3.3(c) shows the velocity vectors on $j=11$ grid surface. The vector is obtained by the calculation of the gradient of potential ϕ. The Fortran program made for solving this problem is uploaded at the Web page:

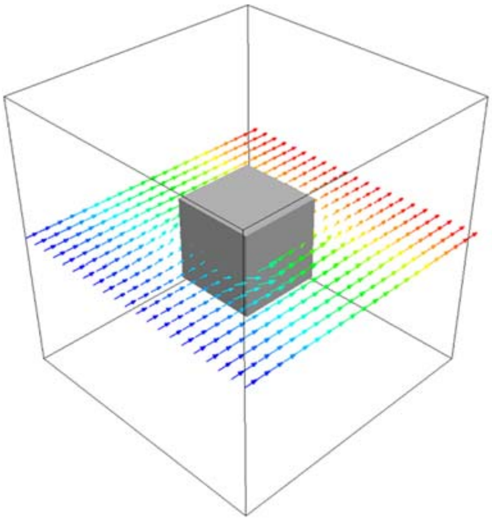
http://www.caero.mech.tohoku.ac.jp/publicData/Numerical/index.html



(a) Computational grid and boundary conditions



(b) Potential distribution



(c) Velocity vectors

Fig. 3.3 Solution of 3-D potential flow

3-3 FDM for Equation of Heat Conduction

1-D equation of heat conduction is written here again:

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2} \tag{3.12}$$

This equation is approximated by the second-order central difference in space and by the first-order forward difference in time as follows:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \kappa \frac{u_{j+1}^{\ n} - 2u_j^n + u_{j-1}^{\ n}}{(\Delta x)^2} \tag{3.13}$$

where u_j^n is the unknown variable *u* at *n* time-step in time direction and at *j* grid point in *x* direction. The assignment expression is derived as

$$u_j^{n+1} = u_j^n + \lambda\left(u_{j+1}^{\ n} - 2u_j^n + u_{j-1}^n\right) \tag{3.14}$$

where $\lambda = \kappa\Delta t/(\Delta x)^2$. FDM employed in Eq. (3.13) is called 'First-order Explicit Euler Forward Method,' where the variable at *n*+1 time-step is calculated from the known variables at *n* time-step only.

As another method, Crank-Nicolson implicit method [8] is known and widely used. Then the 1-D equation is approximated as

$$\frac{u_j^{n+1}-u_j^n}{\Delta t}=\frac{\kappa}{2}\left(\frac{u_{j+1}^{n+1}-2u^{n+1}{}_j+u_{j-1}^{n+1}}{(\Delta x)^2}+\frac{u_{j+1}^{\;n}-2u_j^n+u_{j-1}^{\;n}}{(\Delta x)^2}\right) \tag{3.15}$$

The space difference of $\partial^2 u/\partial x^2$ at n time-step and n+1 time-step are evenly averaged in the equation. The method in which the unknown variables at n+1 time-step both in the left- and right-hand sides are calculated at the same time is called 'Implicit Method'. Solving Eq. (3.15) basically needs the computation of a matrix inversion (see Section 3.5 and 3.6).

3-4 Linear Stability Analysis

As mentioned before, the numerical oscillation when 1-D equation of heat conduction is solved by an explicit method with a large Δt can be explained by the linear stability analysis proposed by von Neumann. The detail may be reported in the paper by Crank and Nicolson [8].

The assignment expression of the first-order explicit Euler forward method applied to the equation of heat conduction is written again as

$$u_j^{n+1}=u_j^n+\lambda\left(u_{j+1}^{\;n}-2u_j^n+u_{j-1}^n\right) \tag{3.16}$$

von Neumann's linear stability analysis assumes that the unknown variable u_j^{n+1} consists of a trigonometric function with an amplitude and arbitrary phases as defined by

$$u_j^n = G^n \exp(ji\theta) \tag{3.17}$$

where $G = G(\theta)$ is the amplitude factor and the function of the phases $\theta=\pi/s$ (s=±1, ±2, ±3, ±4,…). Note that the superscript n of G is not the time-step but the exponentiation. i is the imaginary unit (not the number of grid points). This trigonometric function is substituted into Eq. (3.16). Then we obtain

$$G^{n+1}e^{ji\theta}=G^ne^{ji\theta}+\lambda\left(G^ne^{(j+1)i\theta}-2G^ne^{ji\theta}+G^ne^{(j-1)i\theta}\right) \tag{3.18}$$

Finally we can derive G from Eq. (3.18) as

$$G=1-2\lambda(1-\cos\theta) \tag{3.19}$$

von Neumann's linear stability analysis judges that the FDM form is linearly stable when the absolute value of G is not more than 1. Therefore the stable condition of Eq. (3.19) is obtained when $\lambda \leq 0.5$ as the numerical example in Fig. 2.5(a).

The Crank-Nicolson implicit method applied to 1-D equation of heat conduction is written using λ as

$$-\lambda u_{j+1}^{n+1} + 2(1+\lambda)u_j^{n+1} - \lambda u_{j-1}^{n+1} = \lambda u_{j+1}^{n} + 2(1-\lambda)u_j^{n} + \lambda u_{j-1}^{n} \tag{3.20}$$

The above trigonometric function can be also applied to this form:

$$\begin{aligned} -\lambda G^{n+1}e^{(j+1)i\theta} + 2(1+\lambda)G^{n+1}e^{ji\theta} - \lambda G^{n+1}e^{(j-1)i\theta} \\ = \lambda G^{n}e^{(j+1)i\theta} + 2(1-\lambda)G^{n}e^{ji\theta} + \lambda G^{n}e^{(j-1)i\theta} \end{aligned} \tag{3.21}$$

The value of *G* is finally obtained as

$$G = \frac{1-\lambda(1-\cos\theta)}{1+\lambda(1-\cos\theta)} \tag{3.22}$$

We can find that the absolute value of Eq. (3.22) is always less than one, resulting in unconditionally stable condition. As a result, the Crank-Nicolson method is completely free from the linear stability limitation. However, the value of λ is actually restricted by the boundary conditions, even if the value of λ can sufficiently increase as compared with that of the explicit method.

3-5 Direct Method

The equation of heat conduction approximated by the Crank-Nicolson implicit method is shown here again as

$$-\lambda u_{j-1}^{n+1} + 2(1+\lambda)u_j^{n+1} - \lambda u_{j+1}^{n+1} = \lambda u_{j-1}^{n} + 2(1-\lambda)u_j^{n} + \lambda u_{j+1}^{n} \tag{3.23}$$

where the right-hand side is composed of only known values. If all known values are replaced by simple characters a, b, c and α_j, then we get the simplified equation:

$$a u_{j-1}^{n+1} + b u_j^{n+1} + c u_{j+1}^{n+1} = \alpha_j \tag{3.24}$$

Actually, this equation is defined at each grid point *j* like the series of the equations:

$$\begin{gathered} \cdot \\ \cdot \\ \cdot \\ a u_{j-2}^{n+1} + b u_{j-1}^{n+1} + c u_j^{n+1} = \alpha_{j-1} \\ a u_{j-1}^{n+1} + b u_j^{n+1} + c u_{j+1}^{n+1} = \alpha_j \\ a u_j^{n+1} + b u_{j+1}^{n+1} + c u_{j+2}^{n+1} = \alpha_{j+1} \\ \cdot \end{gathered}$$

·

·

These equations can be written using a matrix and vectors as

$$\begin{bmatrix} \mathrm{O} & & & & & 0 \\ & a & b & c & & \\ & & a & b & c & \\ & & & a & b & c \\ 0 & & & & & \mathrm{O} \end{bmatrix} \begin{bmatrix} \mathrm{M} \\ u_{j-1}^{n+1} \\ u_{j}^{n+1} \\ u_{j+1}^{n+1} \\ \mathrm{M} \end{bmatrix} = \begin{bmatrix} \mathrm{M} \\ \alpha_{j-1} \\ \alpha_{j} \\ \alpha_{j+1} \\ \mathrm{M} \end{bmatrix} \tag{3.25}$$

This matrix form indicates that the vectors of u^{n+1} can be obtained by the matrix inversion. For example, we know the Gauss elimination method and the lower-upper decomposition method as the method solving the matrix inversion. Such methods are called ‘Direct Method’. We only have to calculate the matrix inversion to get u^{n+1}. 1-D equation of heat conduction approximated by the Crank-Nicholson implicit method can be solved by the direct method. However, such a matrix for a multi-dimensional equation is basically quite complicated, and also the computation of a very-large matrix inversion using a direct method may spend a large storage of memories. Direct methods are not useful for large-scale multi-dimensional problems.

3-6 Relaxation Method for Equation of Heat Conduction

Let’s show again the FDM form for 1-D equation of heat conduction approximated by the Crank-Nicolson implicit method:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{\kappa}{2}\left(\frac{u_{j+1}^{n+1} - 2u^{n+1}{}_j + u_{j-1}^{n+1}}{(\Delta x)^2} + \frac{u_{j+1}{}^{n} - 2u_j^n + u_{j-1}{}^{n}}{(\Delta x)^2} \right) \tag{3.26}$$

When $\lambda = \kappa \Delta t / (\Delta x)^2$, then

$$u_j^{n+1} = \lambda \left(u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1} \right) / 2 + \alpha_j \tag{3.27}$$

where

$$\alpha_j = u_j^n + \lambda \left(u_{j-1}^n - 2u_j^n + u_{j+1}^n \right) / 2$$

The solution of Laplace equation at *n*+1 time-step have no meaning until the steady state solution is obtained, while equations of heat conduction are timely dependent and give an accurate solution at each time-step within the finite-difference accuracy. An additional inner iteration like the relaxation methods for Laplace equation is necessary if the same method is applied to Eq. (3.27).

First, if the Jacobi method is applied to Eq. (3.27), we obtain the following equation:

$$\left(u_j^{n+1}\right)^{m+1} = \frac{\lambda}{2(1+\lambda)}\left[\left(u_{j-1}^{n+1}\right)^m + \left(u_{j+1}^{n+1}\right)^m\right] + \frac{\alpha_j}{1+\lambda} \tag{3.28}$$

where *m* is the inner iteration for the relaxation process. While increasing in the iteration number *m* at each *n* time-step, u_j^{n+1} is obtained when the value at *m*+1 iteration is almost equal to that at *m* iteration.

Gauss-Seidel method can be also applied to Eq. (3.28) as follows:

$$\left(u_j^{n+1}\right)^{m+1} = \frac{\lambda}{2(1+\lambda)}\left[\left(u_{j-1}^{n+1}\right)^{m+1} + \left(u_{j+1}^{n+1}\right)^m\right] + \frac{\alpha_j}{1+\lambda} \tag{3.29}$$

Further, Eq. (3.29) is extended to that for SOR method [7]:

$$\left(u_j^{n+1}\right)^{m+1} = \omega\left\{\frac{\lambda}{2(1+\lambda)}\left[\left(u_{j-1}^{n+1}\right)^{m+1} + \left(u_{j+1}^{n+1}\right)^m\right] + \frac{\alpha_j}{1+\lambda}\right\} + (1-\omega)\left(u_j^{n+1}\right)^m \tag{3.30}$$

3-7 Excel Macro Program for 2-D Equation of Heat Conduction

Let's introduce a macro program based on Excel sheet for simulating 2-D equation of heat conduction.

The 2-D equation of heat conduction is defined by

$$\frac{\partial u}{\partial t} = \kappa\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{3.30}$$

As an easy algorithm for Excel macro program, the first-order explicit Euler forward method is employed as FDM:

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} = \kappa\left(\frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{(\Delta x)^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{(\Delta y)^2}\right) \tag{3.31}$$

Setting $\Delta x = \Delta y$, this FDM form is finally reduced to a simple assignment statement:

$$u_{i,j}^{n+1} = u_{i,j}^n + \lambda\left(u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - 4u_{i,j}^n\right) \tag{3.32}$$

Figure 3.4 shows the Excel sheet prepared for initial values. The sheet is composed of Area 1 for setting initial values, Area 2 for storing calculated results at *n* time-step, Area 3 for the calculation at n + 1 time-step and for storing calculation results, and Area 4 for visualizing calculated results. Four variables of Δt, Δx, κ and the time t are fixed in Area 1. Values of the n time-step at the grid points are shown in the Area 2, and the number of grid points is determined as IF= 11 and JF = 11 in this Excel sheet. Cells from C7 to C 17 are entered with 1 as the initial value, and all of other cells are set as 0. The Area 3 is the main part where the calculation at n + 1 time-step is executed and Eq. (3.32) is solved in all cells except ones on the boundary. When the cell D22 is selected, the following arithmetic expression is shown there:

fx = D8+$B$4*$D$4*(E8+C8+D9+D7-D8*4)/$C$4/$C$4

where the $ sign means a cell reference. When $ sign is attached to both a column letter and a line number of a cell, the cell is a fixed reference, that is, each $B$4, $C$4 and $D$4 refers respectively to 0.25, 0.1 and 0.01 which are values of Delta t, Delta x and Kappa in the Area 1. Also, each cell of D8, E8, C8, D9 and D7 respectively correspond to cells in the Area 2 where values of $u_{i,j}^{n}$, $u_{i+1,j}^{n}$, $u_{i-1,j}^{n}$, $u_{i,j+1}^{n}$ and $u_{i,j-1}^{n}$ are stored. Therefore, it is understandable that Eq. (3.32) at the grid point of i = 2 and j = 2 is calculated at the cell of D22. The calculation formula at each grid point, except grid points on boundaries, is automatically formed by copying the calculation formula at D22 into it. On the other hand, concerning boundaries, a calculation of the formula is required at each grid point corresponding to the top and bottom sides given the Neumann boundary condition. Because the calculation formula applying the difference approximation is an explicit solution, the value at each grid point on the boundaries can be easily obtained by extrapolating the value at the neighboring grid point to the boundary which was calculated at n + 1 time-step. For example, the calculation formula of the cell of D21 located on the top side boundary is expressed with *fx* = D22. The procedure for other grid points is the same. Results of the calculation at n + 1 time-step are stored in the Area 3 just after the calculation is finished.

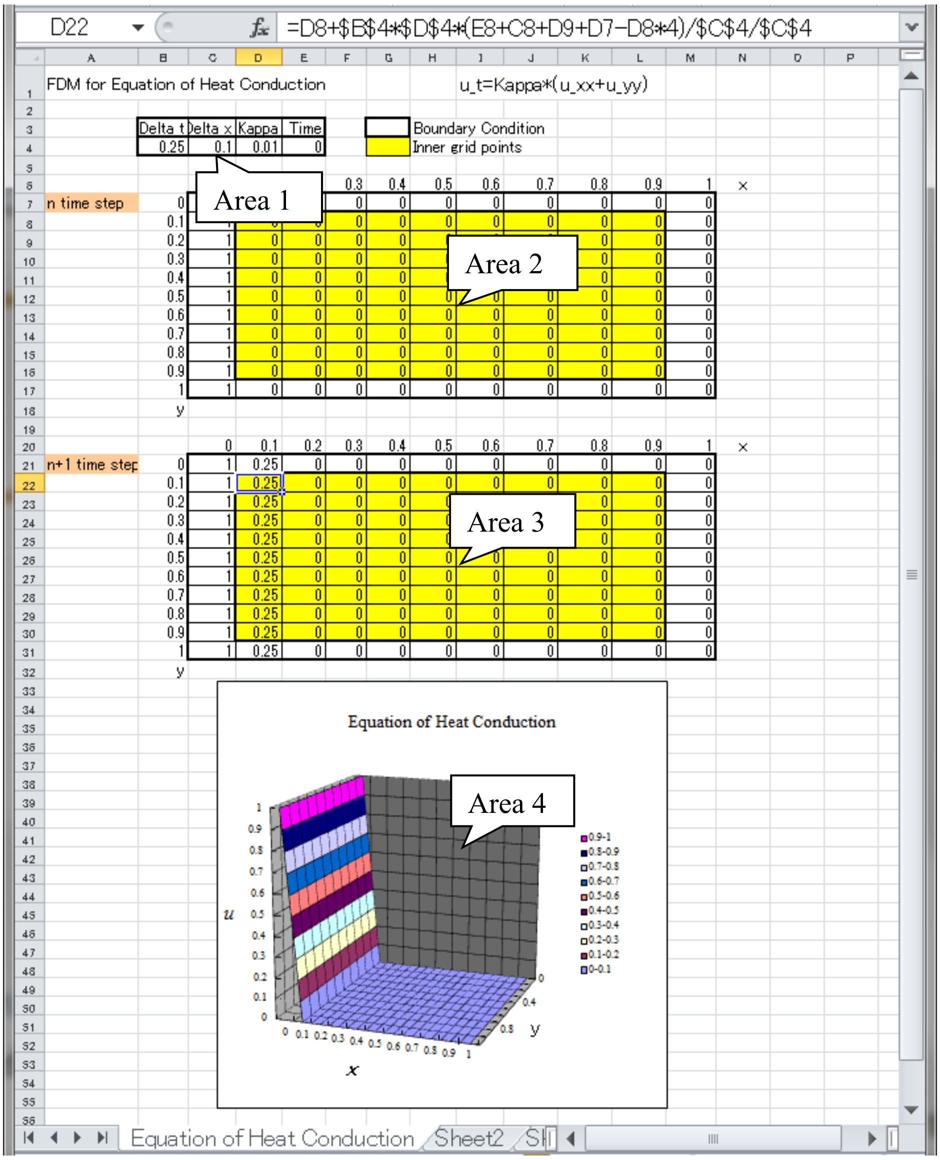
Figure 3.5 shows the sheer for the macro program proceeding the iteration. The work for the calculation is simple because results calculated in Area 3 are merely copied into Area 2 at every time-step, and the routine renewing a value of time is added. As the result, only 7 lines are necessary to make the macro program in Fig. 3.5.

Finally, the name of the macro program is selected by clicking on Tool - Macro – Macro (depends on Windows version), and the macro program is started by clicking on the run button. By the way, the macro program is set to calculate 40 time-steps repeatedly copying values of C21:M31 in Area 3 being a two dimensional cell area to C7:M17 in Area 2 being a two dimensional cell area, and at the same time, the value of Delta t in the cell B4 is added to the Time in the cell E4. The total value of the Time is 0.25 × 40 = 10.0. When there is a need to increase the number of time-steps, the value of a time-step is increased. or the macro program is run repeatedly.

Figure 3.6 shows sheet describing the calculated results in the case of 40 time-steps. The value of the time is renewed and results of the calculation appear in cells of both Area 2 and Area 3 and the results in Area 3 are visualized in Area 4.

This Excel macro program is uploaded at the Web page:

http://www.caero.mech.tohoku.ac.jp/publicData/Modeling/index.html



D22 =D8+$B$4*$D$4*(E8+C8+D9+D7−D8*4)/$C$4/$C$4

FDM for Equation of Heat Conduction u_t=Kappa*(u_xx+u_yy)

Delta t	Delta x	Kappa	Time
0.25	0.1	0.01	0

□ Boundary Condition
🟨 Inner grid points

Area 1

Area 2

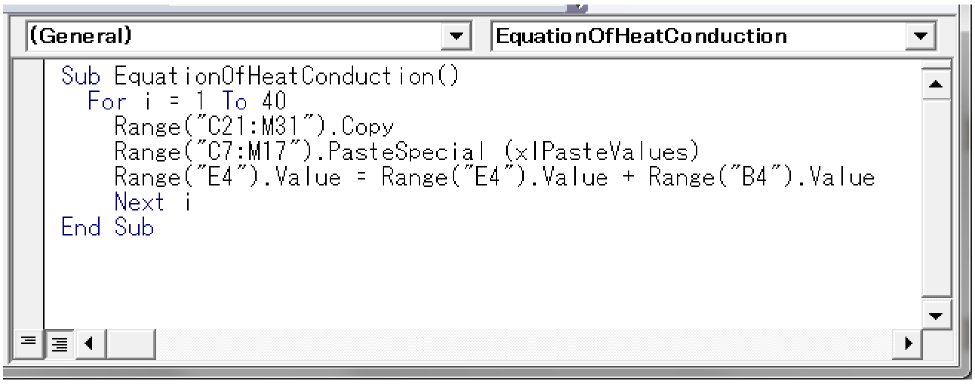
n time step

					0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	x
n time step	0				0	0	0	0	0	0	0	0	
	0.1	1	0	0	0	0	0	0	0	0	0	0	
	0.2	1	0	0	0	0				0	0	0	
	0.3	1	0	0	0	0				0	0	0	
	0.4	1	0	0	0	0				0	0	0	
	0.5	1	0	0	0	0	0	0	0	0	0	0	
	0.6	1	0	0	0	0	0	0	0	0	0	0	
	0.7	1	0	0	0	0	0	0	0	0	0	0	
	0.8	1	0	0	0	0	0	0	0	0	0	0	
	0.9	1	0	0	0	0	0	0	0	0	0	0	
	1	1	0	0	0	0	0	0	0	0	0	0	
	y												

Area 3

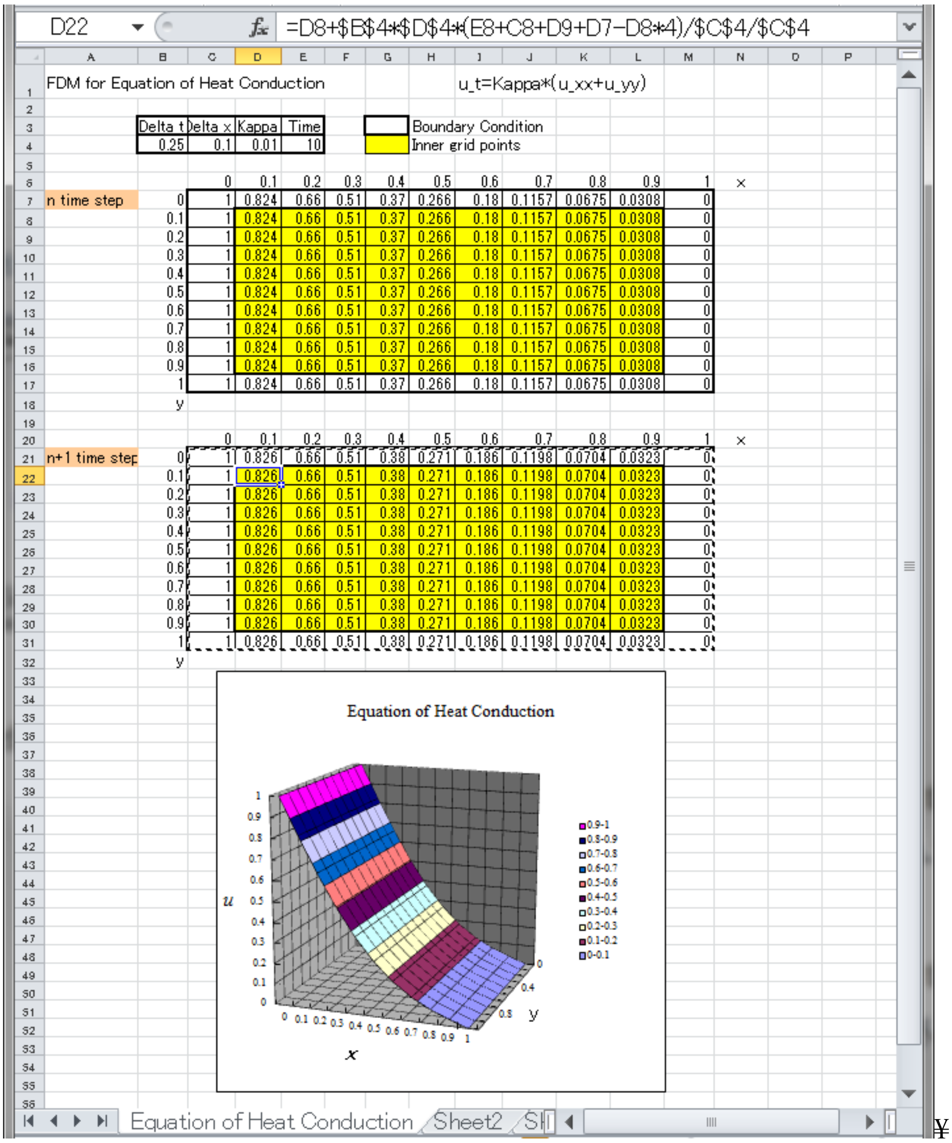
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	x
n+1 time step	0	1	0.25	0	0	0	0	0	0	0	0	0	
	0.1	1	0.25	0	0	0	0	0	0	0	0	0	
	0.2	1	0.25	0	0	0				0	0	0	
	0.3	1	0.25	0	0	0				0	0	0	
	0.4	1	0.25	0	0	0				0	0	0	
	0.5	1	0.25	0	0	0	0	0	0	0	0	0	
	0.6	1	0.25	0	0	0	0	0	0	0	0	0	
	0.7	1	0.25	0	0	0	0	0	0	0	0	0	
	0.8	1	0.25	0	0	0	0	0	0	0	0	0	
	0.9	1	0.25	0	0	0	0	0	0	0	0	0	
	1	1	0.25	0	0	0	0	0	0	0	0	0	
	y												

Fig. 3.4 Excel sheet setting the initial values



```
Sub EquationOfHeatConduction()
  For i = 1 To 40
    Range("C21:M31").Copy
    Range("C7:M17").PasteSpecial (xlPasteValues)
    Range("E4").Value = Range("E4").Value + Range("B4").Value
    Next i
End Sub
```

Fig. 3.5 Macro program



```
=D8+$B$4*$D$4*(E8+C8+D9+D7-D8*4)/$C$4/$C$4
```

FDM for Equation of Heat Conduction

u_t=Kappa*(u_xx+u_yy)

Delta t	Delta x	Kappa	Time
0.25	0.1	0.01	10

□ Boundary Condition

🟨 Inner grid points

n time step

y \ x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0	1	0.824	0.66	0.51	0.37	0.266	0.18	0.1157	0.0675	0.0308	0
0.1	1	0.824	0.66	0.51	0.37	0.266	0.18	0.1157	0.0675	0.0308	0
0.2	1	0.824	0.66	0.51	0.37	0.266	0.18	0.1157	0.0675	0.0308	0
0.3	1	0.824	0.66	0.51	0.37	0.266	0.18	0.1157	0.0675	0.0308	0
0.4	1	0.824	0.66	0.51	0.37	0.266	0.18	0.1157	0.0675	0.0308	0
0.5	1	0.824	0.66	0.51	0.37	0.266	0.18	0.1157	0.0675	0.0308	0
0.6	1	0.824	0.66	0.51	0.37	0.266	0.18	0.1157	0.0675	0.0308	0
0.7	1	0.824	0.66	0.51	0.37	0.266	0.18	0.1157	0.0675	0.0308	0
0.8	1	0.824	0.66	0.51	0.37	0.266	0.18	0.1157	0.0675	0.0308	0
0.9	1	0.824	0.66	0.51	0.37	0.266	0.18	0.1157	0.0675	0.0308	0
1	1	0.824	0.66	0.51	0.37	0.266	0.18	0.1157	0.0675	0.0308	0

n+1 time step

y \ x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0	1	0.826	0.66	0.51	0.38	0.271	0.186	0.1198	0.0704	0.0323	0
0.1	1	0.826	0.66	0.51	0.38	0.271	0.186	0.1198	0.0704	0.0323	0
0.2	1	0.826	0.66	0.51	0.38	0.271	0.186	0.1198	0.0704	0.0323	0
0.3	1	0.826	0.66	0.51	0.38	0.271	0.186	0.1198	0.0704	0.0323	0
0.4	1	0.826	0.66	0.51	0.38	0.271	0.186	0.1198	0.0704	0.0323	0
0.5	1	0.826	0.66	0.51	0.38	0.271	0.186	0.1198	0.0704	0.0323	0
0.6	1	0.826	0.66	0.51	0.38	0.271	0.186	0.1198	0.0704	0.0323	0
0.7	1	0.826	0.66	0.51	0.38	0.271	0.186	0.1198	0.0704	0.0323	0
0.8	1	0.826	0.66	0.51	0.38	0.271	0.186	0.1198	0.0704	0.0323	0
0.9	1	0.826	0.66	0.51	0.38	0.271	0.186	0.1198	0.0704	0.0323	0
1	1	0.826	0.66	0.51	0.38	0.271	0.186	0.1198	0.0704	0.0323	0

¥

Fig. 3.6 Excel sheet representing a time-dependent solution.

3-8 3-D Heat Conduction over A Cubic Body

Next let's solve 3-D equation of heat conduction by the Crank-Nicolson implicit method and SOR method. The fundamental equation and the methods are briefly introduced here.

The fundamental equation considering anisotropic heat conduction is defined by

$$\frac{\partial T}{\partial t}=\frac{\partial}{\partial x}\left(\kappa_x\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(\kappa_y\frac{\partial T}{\partial y}\right)+\frac{\partial}{\partial z}\left(\kappa_z\frac{\partial T}{\partial z}\right) \tag{3.33}$$

where T is temperature. κ_x, κ_y and κ_z are the heat conductivity coefficients in x, y and z directions. This equation is approximated by the Crank-Nicolson implicit method as follows:

$$\begin{aligned}
&\frac{T^{n+1}_{i,j,k}-T^{n}_{i,j,k}}{\Delta t}\\
&=\frac{\kappa_{xr}\left(T^{n+1}_{i+1,j,k}-T^{n+1}_{i,j,k}\right)-\kappa_{xl}\left(T^{n+1}_{i,j,k}-T^{n+1}_{i-1,j,k}\right)}{2(\Delta x)^2}+\frac{\kappa_{xr}\left(T^{n}_{i+1,j,k}-T^{n}_{i,j,k}\right)-\kappa_{xl}\left(T^{n}_{i,j,k}-T^{n}_{i-1,j,k}\right)}{2(\Delta x)^2}\\
&+\frac{\kappa_{yr}\left(T^{n+1}_{i,j+1,k}-T^{n+1}_{i,j,k}\right)-\kappa_{yl}\left(T^{n+1}_{i,j,k}-T^{n+1}_{i,j-1,k}\right)}{2(\Delta y)^2}+\frac{\kappa_{yr}\left(T^{n}_{i,j+1,k}-T^{n}_{i,j,k}\right)-\kappa_{yl}\left(T^{n}_{i,j,k}-T^{n}_{i,j-1,k}\right)}{2(\Delta y)^2}\\
&+\frac{\kappa_{zr}\left(T^{n+1}_{i,j,k+1}-T^{n+1}_{i,j,k}\right)-\kappa_{zl}\left(T^{n+1}_{i,j,k}-T^{n+1}_{i,j,k-1}\right)}{2(\Delta z)^2}+\frac{\kappa_{zr}\left(T^{n}_{i,j,k+1}-T^{n}_{i,j,k}\right)-\kappa_{zl}\left(T^{n}_{i,j,k}-T^{n}_{i,j,k-1}\right)}{2(\Delta z)^2}
\end{aligned} \tag{3.34}$$

where i, j and k are the grid numbers in x, y and z directions. The subscripts r and l for the heat conductivity coefficients indicate for example $\kappa_{xr}=\left[(\kappa_x)_{i+1,j,k}+(\kappa_x)_{i,j,k}\right]/2$ and $\kappa_{xl}=\left[(\kappa_x)_{i,j,k}+(\kappa_x)_{i-1,j,k}\right]/2$ in x direction. Those in other directions are also defined by the same manner. Eq. (3.34) is modified by SOR method and the following assignment form is derived:

$$\begin{aligned}
&\left(T^{n+1}_{i,j,k}\right)^{m+1}=\left(T^{n+1}_{i,j,k}\right)^{m}\\
&+\omega\left[\frac{l_x}{2L}\left\{\kappa_{xr}\left(T^{n+1}_{i+1,j,k}\right)^{m}+\kappa_{xl}\left(T^{n+1}_{i-1,j,k}\right)^{m+1}\right\}+\frac{l_y}{2L}\left\{\kappa_{yr}\left(T^{n+1}_{i,j+1,k}\right)^{m}+\kappa_{yl}\left(T^{n+1}_{i,j-1,k}\right)^{m+1}\right\}\right.\\
&\left.+\frac{l_z}{2L}\left\{\kappa_{zr}\left(T^{n+1}_{i,j,k+1}\right)^{m}+\kappa_{zl}\left(T^{n+1}_{i,j,k-1}\right)^{m+1}\right\}-\left(T^{n+1}_{i,j,k}\right)^{m}+\frac{1}{L}\alpha_{i,j,k}\right]
\end{aligned} \tag{3.35}$$

where

$$\begin{aligned}\alpha_{i,j,k} = T^n_{i,j,k} &+ \frac{l_x}{2}\left\{\kappa_{xr}\left(T^n_{i+1,j,k} - T^n_{i,j,k}\right) - \kappa_{xl}\left(T^n_{i,j,k} - T^n_{i-1,j,k}\right)\right\} \\ &+ \frac{l_y}{2}\left\{\kappa_{yr}\left(T^n_{i,j+1,k} - T^n_{i,j,k}\right) - \kappa_{yl}\left(T^n_{i,j,k} - T^n_{i,j-1,k}\right)\right\} \\ &+ \frac{l_z}{2}\left\{\kappa_{zr}\left(T^n_{i,j,k+1} - T^n_{i,j,k}\right) - \kappa_{zl}\left(T^n_{i,j,k} - T^n_{i,j,k-1}\right)\right\}\end{aligned}$$

$$l_x = \Delta t/(\Delta x)^2,\ l_y = \Delta t/(\Delta y)^2 \text{ and } l_z = \Delta t/(\Delta z)^2$$

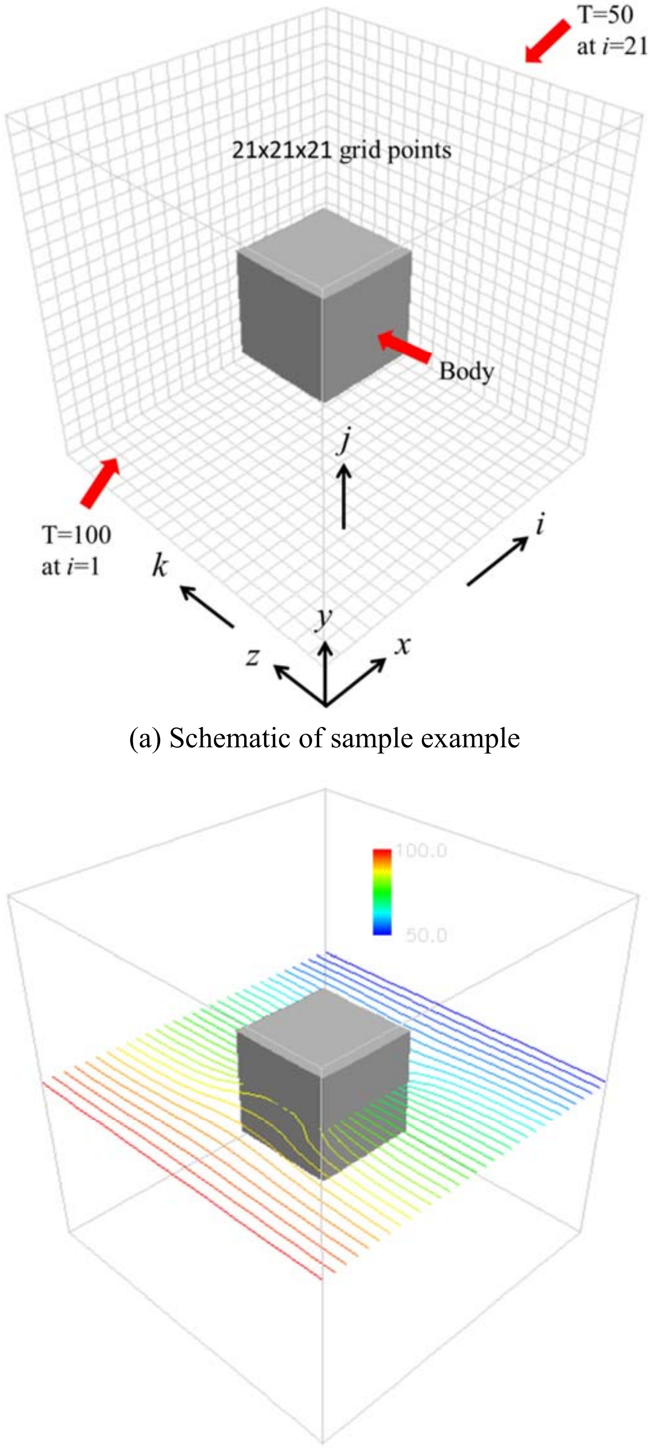
$$L = 1 + \frac{1}{2}l_x(k_{xr} + k_{xl}) + \frac{1}{2}l_y(k_{yr} + k_{yl}) + \frac{1}{2}l_z(k_{zr} + k_{zl})$$

As a sample example for this problem, let's simulate heat conduction over a cubic body shown in Fig. 3.7(a). A rectangular 3-D mesh with 21×21×21 grid points are generated. As the initial condition, the space except for the body is fulfilled with a substance warmed at 50℃. As boundary conditions, the temperature at $i = 1$ boundary surface is heated and fixed to 100℃. The temperature at $i = 21$ boundary surface is only fixed at 50℃. Other boundaries are adiabatic on the temperature and the Neumann boundary condition is applied. The heat conductivity coefficient for the cubic body and the surrounding substance are specified to 1.0 and 5.0.

Figure 3.7(b) shows the calculated temperature at $j = 11$ grid surface obtained after 1000 iterations setting $\Delta t = 0.001$. The temperature is smoothly distributed over the cubic from the inlet to outlet.

This program may be applied to the simulation of reaction-diffusion equations introduced at the following chapter. The Fortran program is uploaded at the Web page:

http://www.caero.mech.tohoku.ac.jp/publicData/Numerical/index.html



(a) Schematic of sample example

(b) Calculated temperature distribution

Fig. 3.7　3-D heat conduction over a cubic body

3-9 Characteristic Equation

A quasi-linear second PDE for unknown variable $u(x, y)$ with respect to two independent variables x, y in a general form is shown again:

$$A\frac{\partial^2 u}{\partial x^2}+2B\frac{\partial^2 u}{\partial x\partial y}+C\frac{\partial^2 u}{\partial y^2}=f\left(x,y,u,\frac{\partial u}{\partial x},\frac{\partial u}{\partial y}\right) \tag{3.36}$$

where the coefficients *A*, *B* and *C* are constants, and *f* at right-hand side is at most a function of *x*, *y* and first partial derivatives.

Defining $D = B^2 - AC$, this equation is categorized to three types according to the sign of *D*:

Hyperbolic	$(D > 0)$
Parabolic	$(D = 0)$
Elliptic	$(D < 0)$

The reason is explained by the theory of characteristics. Then the partial derivatives in Eq. (3.36) are replaced by alphabetic characters as follows:

$$\frac{\partial u}{\partial x}=p,\ \frac{\partial u}{\partial y}=q,\ \frac{\partial^2 u}{\partial x^2}=r,\ \frac{\partial^2 u}{\partial x\partial y}=s,\ \frac{\partial^2 u}{\partial y^2}=t \tag{3.37}$$

Eq. (3.36) is simplified using the characters in Eq. (3.37) as

$$Ar + 2Bs + Ct = f \tag{3.38}$$

The total differentials of p and q are defined by the following equations:

$$dp=\frac{\partial p}{\partial x}dx+\frac{\partial p}{\partial y}dy=rdx+sdy$$

$$dq=\frac{\partial q}{\partial x}dx+\frac{\partial q}{\partial y}dy=sdx+tdy \tag{3.39}$$

Deriving r and s from Eq. (3.39), these values are substituted into Eq. (3.38) as

$$A\frac{dp-sdy}{dx}+2Bs+c\frac{dq-sdx}{dy}=f \tag{3.40}$$

Since the left hand side of Eq. (3.40) is composed of the terms with and without s, we obtain the following equation:

$$s\left\{A\left(\frac{dy}{dx}\right)^2-2B\left(\frac{dy}{dx}\right)+c\right\}-\left\{A\frac{dp}{dx}\frac{dy}{dx}+C\frac{dq}{dx}-f\frac{dy}{dx}\right\}=0 \tag{3.41}$$

This equation should be satisfied for arbitrary s. It is equivalent to satisfy the following two equations:

$$A\left(\frac{dy}{dx}\right)^2 - 2B\frac{dy}{dx} + C = 0 \tag{3.42}$$

and

$$A\frac{dp}{dx}\frac{dy}{dx} + C\frac{dq}{dx} - f\frac{dy}{dx} = 0 \tag{3.43}$$

The former equation Eq. (3.42) is called 'Characteristic Equation'. Eq. (3.42) forms a quadratic equation of the roots obtained as

$$\frac{dy}{dx} = \frac{B + \sqrt{B^2 - AC}}{A} \tag{3.44}$$

and

$$\frac{dy}{dx} = \frac{B - \sqrt{B^2 - AC}}{A} \tag{3.45}$$

$D = B^2 - AC$ corresponds to the value in the root square. If $D > 0$, $D = 0$, and $D < 0$, then Eq. (3.44) and (3.45) as the roots of characteristic equation are reduced to two real roots, duplicate root, and two imaginary roots, respectively. Finally, Eq. (3.36) as a quasi-linear second PDE with two independent variables are categorized according to the roots of the characteristic equation to hyperbolic, parabolic, and elliptic if Eq. (3.44) and (3.45) are two real roots, duplicate root, and two imaginary roots, respectively.

On the other hand, Eq. (3.43) is rewritten using $\lambda = dy/dx$ as

$$A\lambda\frac{dp}{dx} + C\frac{dq}{dx} - f\lambda = 0 \tag{3.46}$$

Because Eq. (3.46) has only one independent variable, Eq. (3.46) divided from the partial differential equation Eq. (3.36) results in an ordinary differential equation.

Let's explain the roots of characteristic equation geometrically. Fig. 3.8 shows two lines drawn from two points $(x_1,0)$, $(x_2,0)$ on the initial value $y = 0$ $(y > 0)$. If the gradient of each line dy/dx equals to the root of the characteristic equation Eq. (3.42), these lines indicate the trace of solutions of Eq. (3.36) propagated from the two points. Such lines are called 'Characteristic Curves' (note that those are curves in second order accuracy, but lines in first order accuracy).

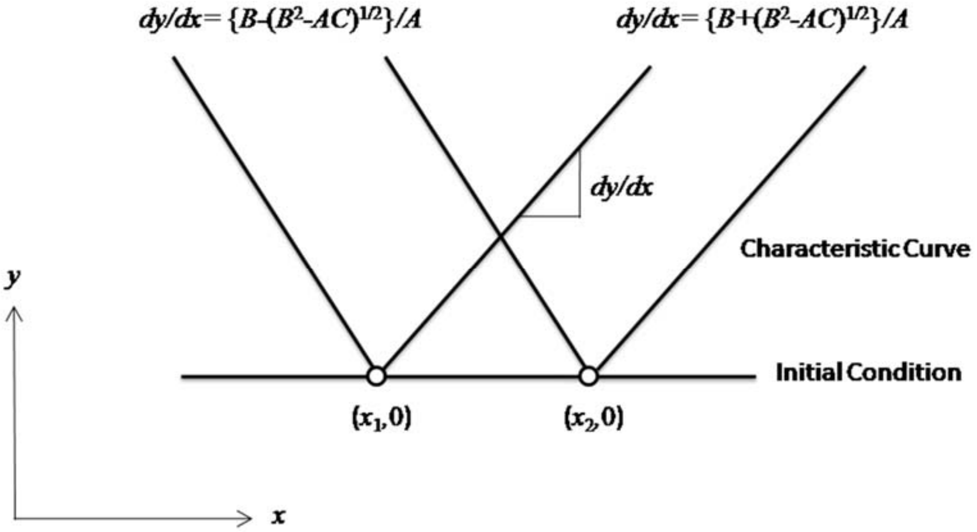


Fig. 3.8 Characteristic curves and the gradient.

Characteristic curves propagated from the points L and M are shown in Fig. 3.9. The roots of the characteristic curves extending to x and $-x$ directions are defined as λ^+ and λ^-, respectively. The curve with the gradient (root) λ_L^+ propagated from the point L satisfies the following ordinary differential equation.

$$A\lambda_L^+ \frac{dp}{dx} + C\frac{dq}{dx} - f\lambda_L^+ = 0 \qquad (3.47)$$

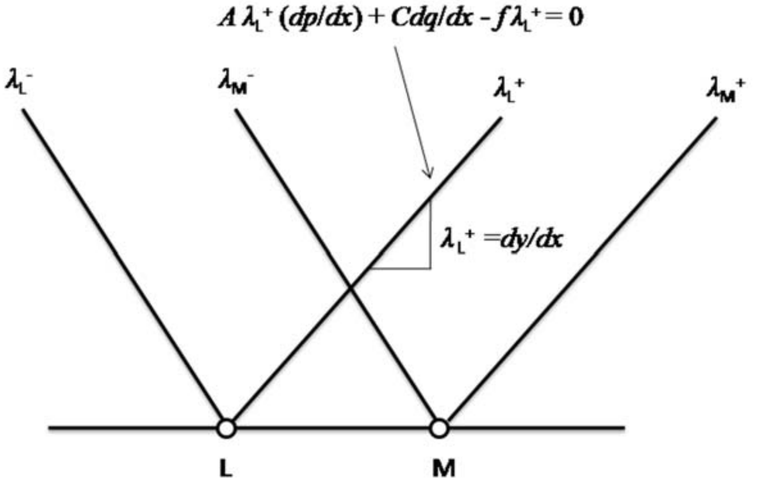


Fig. 3.9 Ordinary differential equation satisfied on characteristic curve.

4. Mathematical Modeling based on Reaction Equations

4-1 SIR Model and the Extension

SIR model proposed by Kermack and McKendrick [1] - [3] was introduced at Chapter 2. This model is a typical well-known system of reaction equations which can predict the transmission of infection. SIR model is composed of three reaction equations as shown again:

$$\begin{aligned} \frac{dS(t)}{dt} &= -\beta\, S(t)I(t) \\ \frac{dI(t)}{dt} &= -\gamma\, I(t) + \beta\, S(t)I(t) \\ \frac{dR(t)}{dt} &= \gamma\, I(t) \end{aligned} \tag{4.1}$$

where $S(t)$ is the density of population under susceptible condition. $I(t)$ is the density of population under infectious condition. $R(t)$ is the density of population under recovered, removed or immune condition. β is the transmission rate from susceptible condition to infectious condition and γ is the recovered rate; both are empirical constants. The total population $N(t) = S(t) + I(t) + R(t)$ is constant and the above equations derives

$$\frac{dS(t)}{dt} + \frac{dI(t)}{dt} + \frac{dR(t)}{dt} = 0 \tag{4.2}$$

A typical result solving Eq. (4.1) was represented at Chapter 2. In this Chapter, Eq. (4.1) is further modified by considering additional physics and events. Eq. (4.1) can simulate the transmission of infection under a constant population. However, actually some events such as the birth, death, immigrant, and the changing age may occur. Eq. (4.1) is modified by considering the birth and death:

$$\begin{aligned} \frac{dS(t)}{dt} &= b - \mu S(t) - \beta\, S(t)I(t) \\ \frac{dI(t)}{dt} &= -(\gamma + \mu)\, I(t) + \beta\, S(t)I(t) \\ \frac{dR(t)}{dt} &= -\mu R(t) + \gamma\, I(t) \end{aligned} \tag{4.3}$$

where b is the birth rate and μ is the normal death rate. This model is called Endemic SIR model. To prevent influenza virus, people may receive the vaccine. The effect of vaccine can be also considered in the model as follows:

$$\begin{aligned}\frac{dS(t)}{dt}&=b(1-e)-\mu S(t)-\beta\,S(t)I(t)\\ \frac{dI(t)}{dt}&=-(\gamma+\mu)\,I(t)+\beta\,S(t)I(t)\\ \frac{dR(t)}{dt}&=be-\mu R(t)+\gamma\,I(t)\end{aligned}\tag{4.4}$$

where e is the inoculation rate of vaccine for children. Population who inoculated the vaccine is moved directly to the population under recovered. Even though the number is small, a part of population may be infected again after the recover from the first infection. Then Eq. (4.4) is modified considering the reinfection as

$$\begin{aligned}\frac{dS(t)}{dt}&=b(1-e)-\mu S(t)-\beta\,S(t)I(t)\\ \frac{dI(t)}{dt}&=-(\gamma+\mu)\,I(t)+\beta\left[S(t)+\sigma R(t)\right]I(t)\\ \frac{dR(t)}{dt}&=be-\mu R(t)+\gamma\,I(t)-\beta\sigma R(t)I(t)\end{aligned}\tag{4.5}$$

where $\beta\sigma$ is the reinfection rate.

4-2 SEIR Model

The symptoms of fever and cough due to the influenza may start with a latent period after catching the virus. The latent period can be modeled by introducing an additional equation. The system of four equations are defined by

$$\begin{aligned}\frac{dS(t)}{dt}&=b-\mu S(t)-\beta\,S(t)I(t)\\ \frac{dE(t)}{dt}&=-(\mu+\varepsilon)E(t)+\beta\,S(t)I(t)\\ \frac{dI(t)}{dt}&=-(\gamma+\mu)\,I(t)+\varepsilon E(t)\\ \frac{dR(t)}{dt}&=-\mu R(t)+\gamma I(t)\end{aligned}\tag{4.6}$$

where E is the population under latent period. ε is the transmission rate from latent period to infection period. This model is called ‘SEIR Model’.

Mathematical models are being updated considering additional physics and events like the developing story from Eq. (4.1) to Eq. (4.6). These models can further include diffusion terms for considering the space distribution of the population like the next Chapter 5, extending to the system of PDE as reaction-diffusion equations.

5. Mathematical Modeling based on Reaction-Diffusion Equations

5-1 Reaction-Diffusion Equation

The equation of heat conduction coupled with a reaction term is called ‘Reaction-Diffusion Equation (RDE)’. Let’s define a RDE as

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2} + au \tag{5.1}$$

where a is a constant. The source term au is easily added to the FDM form in Eq. (3.28) based on the Crank-Nicolson and Jacobi method as

$$\left(u_j^{n+1}\right)^{m+1} = \frac{\lambda}{2(1+\lambda)}\left[\left(u_{j-1}^{n+1}\right)^m + \left(u_{j+1}^{n+1}\right)^m\right] + \frac{\alpha_j}{1+\lambda} + \Delta tau_j^n \tag{5.2}$$

As a well-known RDE, the following RDE with a cubic source term was proposed:

$$\frac{\partial u}{\partial t} = \varepsilon^2 \frac{\partial^2 u}{\partial x^2} + u(u-a)(1-u) \tag{5.3}$$

where ε is an empirical constant. This RDE is called ‘Nagumo’s Equation’ [2] and is a typical mathematical model for simulating a signal transmission of neuro system. Also the following equation is called ‘Fisher’s Equation’ [9]:

$$\frac{\partial u}{\partial t} = \varepsilon^2 \frac{\partial^2 u}{\partial x^2} + au(1-u) \tag{5.4}$$

This RDE can simulate a growth and diffusion in ecological system. In particular, the RDE setting ε = 0 reduces to a reaction equation called ‘Logistic Equation’ which can predict a population increase. Generally, a lot of phenomena and social events may be modeled by RDE.

Now let’s solve the Fisher’s RDE. Setting $0 \leqq x \leqq 1$, ε = 0.001, $a = -1$, $u(x,0)$ = 0.01 + 0.1 sin πx and Δt = 0.1, we obtain a result shown in Fig. 3.1(a) in which u increases while increasing in time and the value is the maximum at x = 0.5. We also obtain another result in Fig. 5.1(b) when a = 1. In contrast to Fig. 3.1(a), u decreases while increasing in time even if the maximum value is located at x = 0.5.

Fisher’s equation can simulate the increase or decrease of u as changing a and the distribution of u as changing the diffusion parameter ε. This equation may apply for simulating the growth and disappearance of the number density of an ecological system if u is defined by the number.

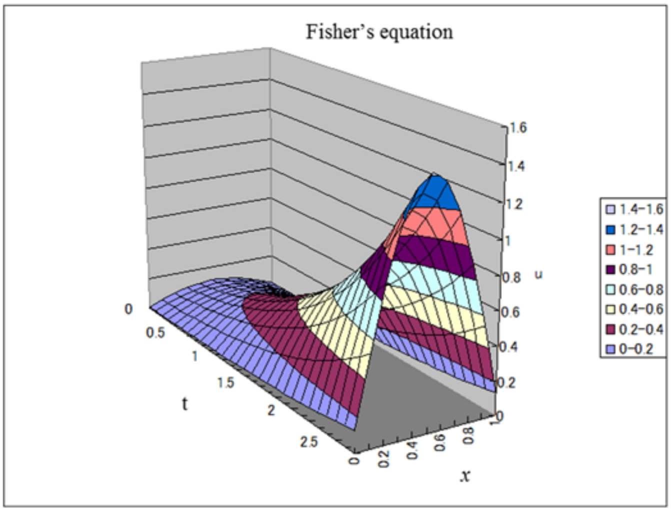


Fig. 5.1(a) Time-dependent solution for Fisher's equation ($a=-1$).

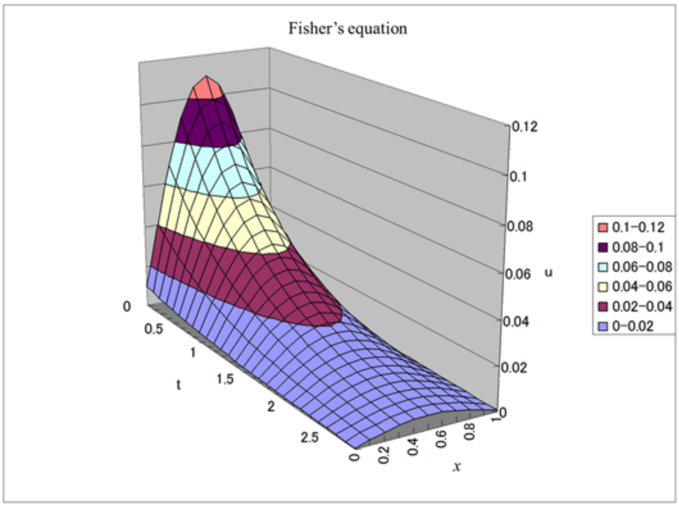


Fig. 5.1(b) Time-dependent solution for Fisher's equation ($a=1$).

5-2 System of RDE's

System of RDEs can simulate a number of more complicated physics and social events. Let's consider the following equations:

$$\begin{aligned}\frac{\partial u}{\partial t}&=\varepsilon^2\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+avu\\ \frac{\partial v}{\partial t}&=\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}-uv\end{aligned} \tag{5.5}$$

where u and v are unknown variables, ε is a diffusion coefficient and a is an empirical constant. Initial values for u and v are specified as shown in Figs. 5.2(a) and 5.2(b), setting

$0 \leqq x \leqq 1$, $0 \leqq y \leqq 1$, $\varepsilon = 0.2$ and $a = 5$. As boundary conditions, $\partial u/\partial n = 0$ and $\partial v/\partial n = 0$ toward the normal direction at all boundaries. As initial values, $u(x,y,0)$=exp[{(x - 0.5)^2 +(y - 0.5)^2}*50] and $v = 1$.

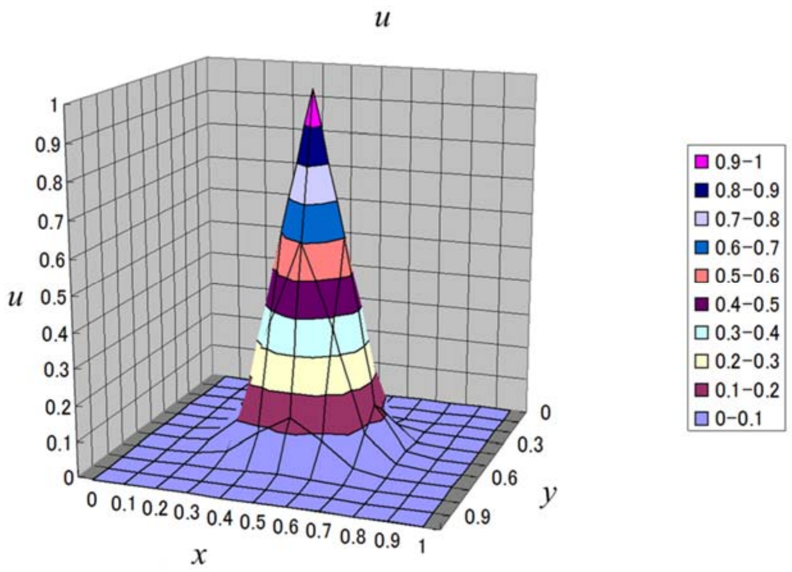


Fig. 5.2(a) Initial value for u .

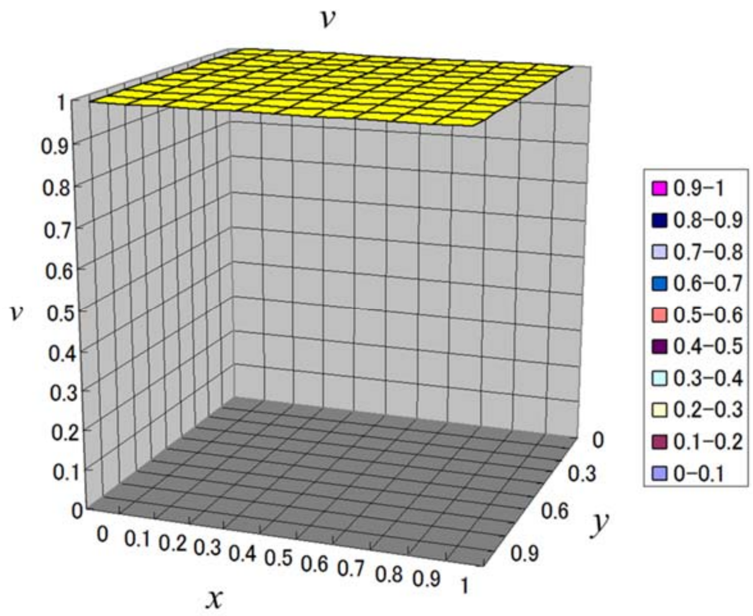


Fig. 5.2(b) Initial value for v .

When $t = 0.3$ after 300 iterations with $\Delta t = 0.001$, we obtain the solutions shown in Figs. 5.2(c) and 5.2(d). We can find that the maximum of u increases 1.5 times more than that of initial value and the distribution is more widely spread, while v decreases and has the minimum value at the same point where u has the maximum value. This system of RDEs may simply simulate for example the growth of bacteria in nutritive system, where u and v are the densities of bacteria and nutritive.

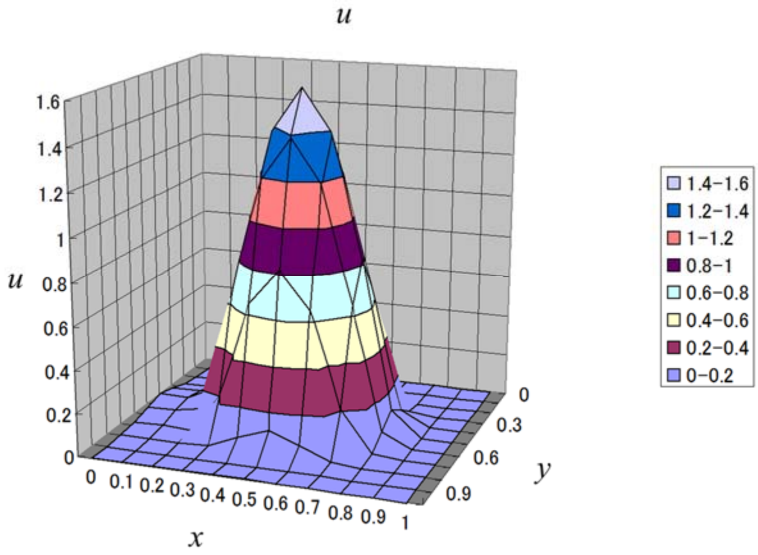


Fig. 5.2(c) Time-dependent solution for u .

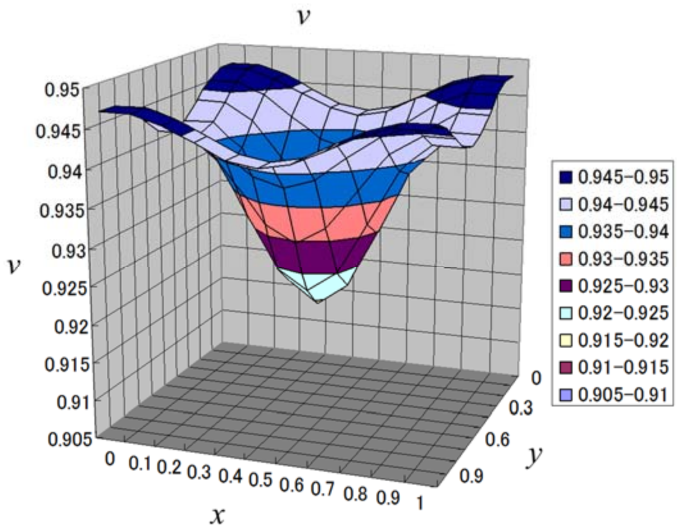


Fig. 5.2(d) Time-dependent solution for v .

A diffusive chemical reaction governing the reaction process

$$\mathrm{A} + \mathrm{B} \rightarrow \mathrm{C} \tag{5.6}$$

is modeled by a system of RDEs, where A and B are reactive substances and C is the product substance. This process is modeled by the following three RDEs:

$$\begin{aligned}
\frac{\partial u}{\partial t} &= D_u \nabla^2 u - avu \\
\frac{\partial v}{\partial t} &= D_v \nabla^2 v - auv \\
\frac{\partial w}{\partial t} &= D_w \nabla^2 w + auv
\end{aligned} \tag{5.7}$$

where u, v and w are the densities of A, B and C. D_u, D_v and D_w are the diffusion coefficients for u, v and w; a is the reaction rate. Note that $\nabla^2 u$ is the diffusion term represented by Nabla operator.

5-3 A Mathematical Model for Bacteria and White Cell

Alt and Lauftenberger [10] proposed a mathematical model which can simulate the defensive system of white cell against bacteria. This system is modeled by the assumption: 1) bacteria is increased and diffused. 2) bacteria excrete an attractant. 3) white cell approaches bacteria according to the density of the attractant. 4) white cell eats bacteria; both bacteria and white cell are broken at the same time. Then the model is written by the following three equations:

$$\begin{aligned}
\frac{\partial u}{\partial t} &= D_u \nabla^2 u + (a - bw)u \\
\frac{\partial v}{\partial t} &= D_v \nabla^2 v + cu \\
\frac{\partial w}{\partial t} &= D_w \nabla^2 w - (d + eu)w - \nabla \cdot (\alpha w \nabla v)
\end{aligned} \tag{5.8}$$

where u is the density of bacteria, v is the density of the attractant and w is the density of white cell. a, b, c, d, e and α are empirical constants. Note that the third term at the right-hand side of third equation is expanded for three-dimension as

$$\begin{aligned}
&\nabla \cdot (\alpha w \nabla v) \\
&= \left(\frac{\partial}{\partial x} \; \frac{\partial}{\partial y} \; \frac{\partial}{\partial z} \right) \cdot \left(\alpha w \frac{\partial v}{\partial x} \;\; \alpha w \frac{\partial v}{\partial y} \;\; \alpha w \frac{\partial v}{\partial z} \right) \\
&= \frac{\partial}{\partial x}\left(\alpha w \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial y}\left(\alpha w \frac{\partial v}{\partial y} \right) + \frac{\partial}{\partial z}\left(\alpha w \frac{\partial v}{\partial z} \right)
\end{aligned}$$

This term approximates the movement of white cell to bacteria according to the density of the attractant. Alt and Lauftenberger assumed one-dimensional problem. Here the above equations are further extended to two-dimensional equations with the consideration of nutritive for bacteria. Then the following four equations are obtained:

$$\begin{aligned}
\frac{\partial u}{\partial t} &= D_u \nabla^2 u + (a - bw)u + \beta nu \\
\frac{\partial v}{\partial t} &= D_v \nabla^2 v + cu \\
\frac{\partial w}{\partial t} &= D_w \nabla^2 w - (d + eu)w - \nabla \cdot (\alpha w \nabla v) \\
\frac{\partial n}{\partial t} &= D_n \nabla^2 n - \beta un
\end{aligned} \tag{5.9}$$

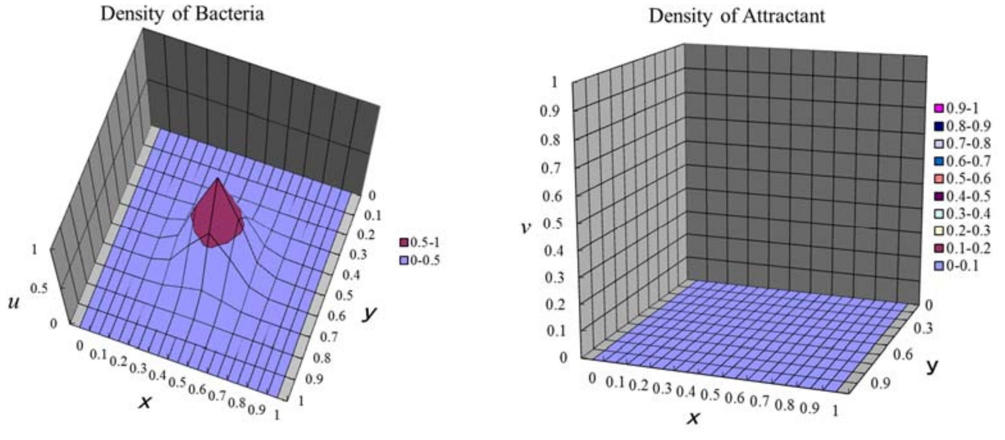
where n is the density of nutritive and β is an empirical constant.

One of typical solutions for Eq. (5.9) is introduced here. Empirical constants set here are summarized in Table 1.

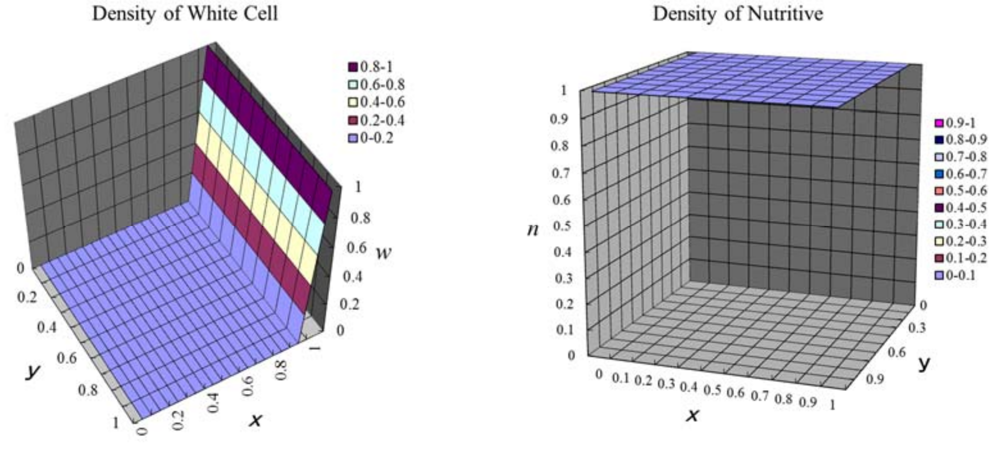
Table 1 Empirical constants in Eq. (5.9)

a	b	c	d	e	α	β
1.0	100.0	1.0	1.0	100.0	1.0	10.0

and $D_u = D_v = D_w = D_n = 1.0$. If initial conditions setting $u(x, y, 0)$=exp[{$(x - 0.5)^2 + (y - 0.5)^2$}*50], $v(x, y, 0) = 0.0$, $w(1, y, 0) = 1.0$ and $n(x, y, 0) = 1.0$ at $0 \leqq x \leqq 1$ and $0 \leqq y \leqq 1$ are visualized in Fig. 5.3.



(a) (b)

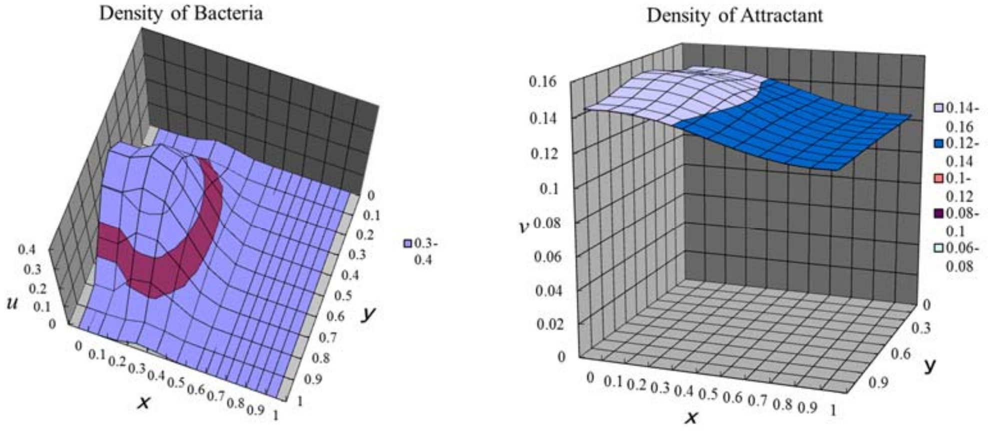


(c) (d)

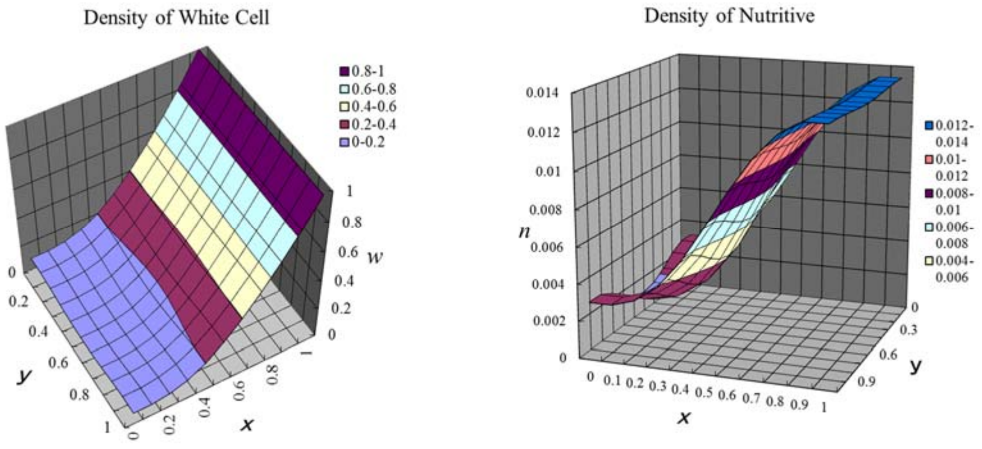
Fig. 5.3 Initial values [(a) density of bacteria, (b) density of attractant, (c) density of white cell (d) density of nutritive].

Figure 5.4 shows the calculated results after 400 time-steps setting $\Delta t = 0.001$. The density of bacteria shown in Fig. 5.4(a) relatively decreases and the maximum location moves toward -*x* direction. The density of attractant is found and the value near the region where the density of bacteria is higher may be also slightly higher than that around the location as shown in Fig. 5.4(b). Fig. 5.4(c) indicates that white cell approaches bacteria. The value may be relatively lower near the region where the density of bacteria is higher. The density of nutritive in Fig. 5.4(d) decreases due to the consumption by the bacteria. From these results, the assumption of the mechanism for bacteria and white cell modeled in Eq. 5.9 may be represented in Fig. 5.4. The Excel macro program for this problem is uploaded at

http://www.caero.mech.tohoku.ac.jp/publicData/Modeling/index.html



(a) (b)



(c) (d)

Fig. 5.4 Time-dependent solutions [(a) density of bacteria, (b) density of attractant, (c) density of white cell (d) density of nutritive].

3-D cases for bacteria and white cell were also simulated by solving 3-D equations of Eq. 5.9. Only a typical visualized result is shown here. Setting all empirical constants properly and $u(x, y, z, 0)=\exp[\{(x - 0.5)^2 + (y - 0.5)^2 \;\; + (z - 0.5)^2\}*50]$, $v(x, y, z, 0) = 0.0$, $w(1, y, z, 0) = 1.0$ and $n(x, y, z, 0) = 1.0$ as initial values at $0 \leqq x \leqq 1$, $0 \leqq y \leqq 1$ and $0 \leqq z \leqq 1$ obtains an early relation between bacteria and white cell as shown in Fig. 5.5(a).

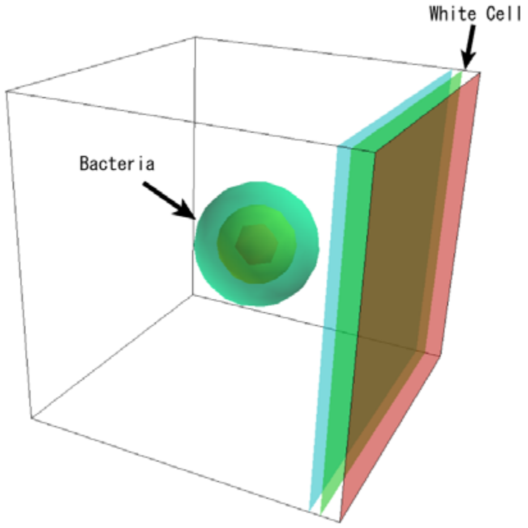


Fig. 5.5(a) An early result showing densities of bacteria and white cell.

After a number of time-steps, the relationship is changed as the following result in Fig. 5.5(b). Although the reliability of solution is not guaranteed, white cell moves toward the bacteria. The density distribution of bacteria was finally forced to be deformed due to the approach of white cell.

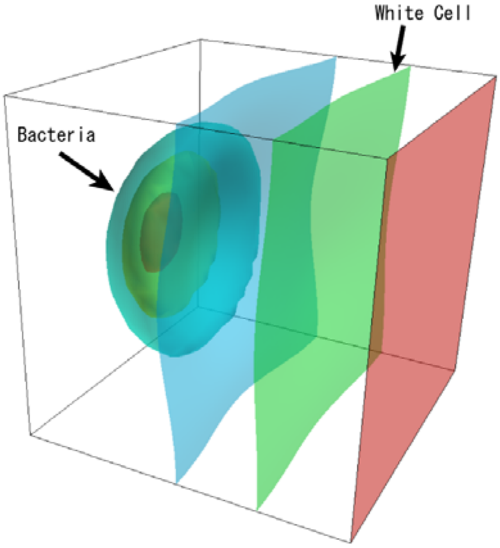


Fig. 5.5(b) A time-dependent result showing densities of bacteria and white cell.

5-4 Mathematical Model for Self-replicating Pattern

More than half century ago, Turing [11] who is well-known as the machine learning also presented a paper proposing that a space non-uniform solution can be obtained from a reaction-diffusion equation. In addition, he argued that cell differentiation and the morphogenesis are governed by diffusion phenomena. Almost at the same time, Hodgkin and Huxley [1] proposed that a system of RDE's can obtain a solution of the pulsative signal of ion going in and out a nerve membrane. After forty years, Pearson [12] presented a paper in which complex pattern can be simulated by a simple system such as RDEs. His computer simulation verified that the arguments by Turing, Hodgkin and Huxley. A unique system of RDEs, let's introduce the so-called Gray-Scott model simulated by Pearson here.

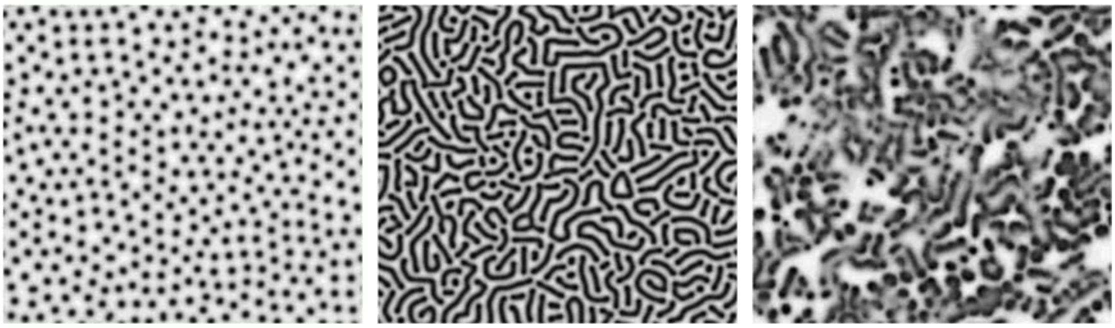
Gray and Scott [13] presented a paper in which an auto catalytic reaction is observed in a gel medium. The system of reactions was reported as

$$\begin{gathered} \mathrm{A} + 2\mathrm{B} \rightarrow 3\mathrm{B} \\ \mathrm{B} \rightarrow \mathrm{C} \end{gathered} \tag{5.10}$$

A, B and C are chemical substances and the substance A works as a catalytic one. The mathematical model governing the reactions is defined by

$$
\begin{aligned}
&\frac{\partial u}{\partial t} = D_u \nabla^2 u - uv^2 + f(1-u) \\
&\frac{\partial v}{\partial t} = D_v \nabla^2 v + uv^2 - (f+k)v \\
&\left(\frac{\partial w}{\partial t} = D_w \nabla^2 w + kv \right)
\end{aligned}
\tag{5.11}
$$

where u, v and w are the densities of substance A, B and C. f and k are empirical constants. Pearson reported that a slight change of the two empirical constants induced a large difference of solution and each solution has its own unique pattern. He categorized solutions into 12 patterns where he sets $D_u = 2\times10^{-5}$ and $D_v = 10^{-5}$. Modifying the Fortran program for 2-D equation of heat conduction with 256×256 grid points, three typical patterns could be obtained. When $f = 0.024$ and $k = 0.063$, a pattern fulfilled with spots which are increased by self-duplication from initial values was obtained as shown in Fig. 5.6(a). Fig. 5.6(b) shows a time-dependent solution like a labyrinth pattern when $f = 0.044$ and $k = 0.063$. Fig. 5.6(c) shows a time-dependent chaotic pattern when $f = 0.02$ and $k = 0.052$.



(a) $f = 0.024$, $k = 0.063$ (b) $f = 0.044$, $k = 0.063$ (c) $f = 0.02$, $k = 0.052$

Fig. 5.6 Time-dependent solutions of Gray-Scott model while changing f and k.

3-D cases have been also simulated by 3-D Gray-Scott model. Following two figures show the time-dependent results at a different time with a same initial and boundary conditions representing a chaotic behavior. 3-D Fortran program for heat conduction was modified for this problem.

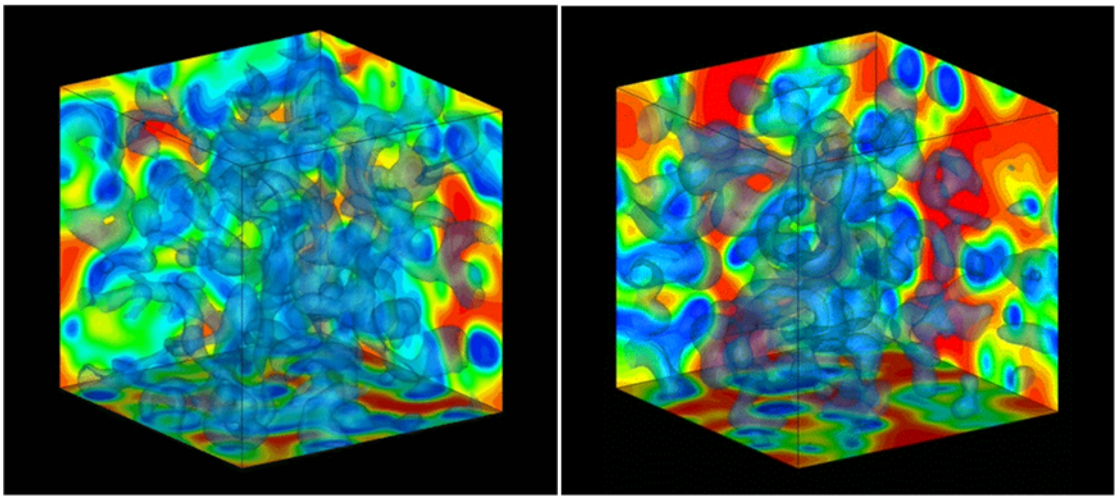


Fig. 5.7 Time-dependent solutions at a different time for 3-D Gray-Scott model.

6. Concluding Remarks

The basis of mathematical modeling based on partial differential equations and the computational method using the finite-difference method were first explained in this lecture note. Next several typical mathematical models formed by reaction equations or reaction-diffusion equations were briefly introduced to assist for understanding how to make a mathematical model. I believe that everybody who understands this lecture can make their own mathematical models soon hereafter.

References:

1. A.L. Hodgkin and A.F. Huxley, A Quantitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerve, Journal of Physiology, 117(1952), 500-544.
2. H.A. Abdusalam, Analytic and Approximate Solutions for Nagumo Telegraph Reaction Diffusion Equation, Applied Mathematics and Computation, 157(2004), 515-522.
3. W.O. Kermack and A.G. McKendrick, Contributions to the Mathematical Theory of Epidemics I, Proc. of the Royal Society, 115A(1927), 700-721, [reprinted in Bulletin of Mathematical Biology, 53(1/2)(1991), 33-55].
4. W.O. Kermack and A.G. McKendrick, Contributions to the Mathematical Theory of Epidemics II, Proc. of the Royal Society, 138A(1932), 55-83, [reprinted in Bulletin of Mathematical Biology, 53(1/2)(1991), 57-87].

5. W.O. Kermack and A.G. McKendrick, Contributions to the Mathematical Theory of Epidemics III, Proc. of the Royal Society, 141A(1933), 94-122, [reprinted in Bulletin of Mathematical Biology, 53(1/2)(1991), 89-118].
6. A. J. Lotka, Elements of Mathematical Biology, Dover, (1956), [reprinted from A. J. Lotka, Elements of Physical Biology, Williams and Wilkins, (1925)].
7. S.P. Frankel, Convergence Rates of Iterative Treatments of Partial Differential Equations, Mathematical Tables and Other Aids to Computation, 4(1950), 65-75.
8. J. Crank and P. Nicholson, A Practical Method for Numerical Evaluation of Solution of Partial Differential Equations of the Heat-Conduction type, Proc. Cambridge Pilos. Soc., 43(1947), 50-67.
9. R. A. Fisher, The Genetical Theory of Natural Selection. Oxford University Press, (1930), [New Ed Edition, (2000)].
10. W. Alt and D.A. Lauftenberger, Transient Behavior of a Chemotaxis System Modelling Certain Types of Tissue Inflammation, Journal of Mathematical Biology, 24(1987), 691-722.
11. A.M. Turing, The Chemical Basis of Morphogenesis, Phil. Trans. Royal Society of London, B327(1952), 37-72.
12. J.E. Pearson, Complex Patterns in A Simple System, Science, 61(1993), 189-192.
13. P. Gray and S.K. Scott, Autocatalytic Reaction in the Isothermal, Continuous, Stirred Tank Reactor: Oscillation and Instabilities in the System $A + 2B \rightarrow 3B$, B to C, Chemical Engineering Science, 39(1984), 1087-1097.