Regular Paper

Graphic Presentation of Chaos Generated in Computational Fluid Dynamics

TOSHIRO ARATANI*

A method is proposed to graphically display the instability in the numerical solution of fluid dynamic equation. The Navier-Stokes equations are modified, based on the assumption that the pressure gradient is proportional to the gradient of kinetic energy. Modified partial differential equations are expressed only in terms of velocity components.

The resulting differential equations are approximated by the finite differences with respect to space variables, where asymmetrical finite difference has been applied to some terms. A set of ordinary differential equations with time are obtained. This set of equations is numerically solved by the Eulerian scheme with a finite time step. The magnitudes of the kinetic energy at every mesh point are displayed on a graphics screen.

The numerical method is applied to a problem corresponding to a two-dimensional unsteady state fluid flow around an obstacle, suddenly put into a uniform fluid flow stream. The number of 200 by 400 mesh points in the discretization are large enough to simulate the vortex shedding behind the obstacle. Chaos can be observed in the wake, owing to the propagation of numerical integration errors, at large time step. The lower limit of the time step for the generation of chaos is determined by numerical experiments.

1. Introduction

Recent advance in computer hardware technology has provided us inexpensive high speed computation with large memory space. The hardware enables us to simulate dynamic systems described by partial differential equations. As an example of the system of continuum, two-dimensional dynamics of the Newtonian fluid is simulated by a computational numerical method developed, as well as by a graphic display method [3, 9].

Numerical solutions of partial differential equation greatly depend on the numerical methods used. For fluid dynamics, many numerical methods have been used. The representatives are finite difference method, finite element method and boundary element method. These methods, however, involve some difficulties as follows [5, 8, 10, 11].

The first difficulty is the large number of divisions required for the space variables. The maximum number is restricted to the upper limit of the memory capacity of the computer available. Highly accurate numerical solution requires sufficiently large number of divisions, especially for the simulation of vortices and eddies or turbulence. The way of divisions in a direction with a thousand for a three-dimensional scalar equation needs $1000 \cdot 1000 \cdot 1000 \cdot 4 = 4$ giga bytes memory spaces, because real number needs a storage consisting of 4 bytes or 32 bits with floating-point arithmetics. The dynamic case requires additional storage for time marching scheme. Euler scheme requires twice as large storage or 8 giga bytes, and Runge-Kutta scheme requires four times as large storage or 16 giga bytes as in the steady case. This way of division is still insufficient to simulate eddies or turbulence with high accuracy. In this work several hundred divisions are used for a two-dimensional fluid dynamics simulation, although the author does not claim that the results are quantitatively real.

The second difficulty is the problem of stability. Stability of numerical solution crucially depends on the finite difference formulae. The author has proposed an explicit asymmetrical finite difference formulae to momentum transport term. In this work the governing equations are derived from mass and momentum balances, through a different manipulation from the derivation of conventional Navier-Stokes equation [2].

The third difficulty consists in the pressure term. The pressure gradient is assumed to be proportional to the difference in the kinetic energy at two neighbouring mesh points.

Streamlines or equipotential lines are usually displayed on a graphics. These lines, however, move with time. The local kinetic energy or the dynamic pressure is computed at each mesh point, and the magnitude is used to select colors at such mesh point. The color level set algorithm was originated by Benoit B. Mandelbrot for his fractal graphics [3].

Numerical instability, due to the discretization of nonlinear dynamic system with finite time step, is

*Department of Management and Information Sciences, Hiroshima Prefectural University, Shyobara-shi, Hiroshima 727, Japan.

known to produce chaotic oscillation. The numerical integrations by iterations to advance time can be a cause for the chaotic oscillation, when the finite time difference is so large as to give numerical instability. Far bigger finite time step leads to the overflow in the floating-point computation. In this case, velocity components are reduced artificially to avoid the numerical overflow.

2. Numerical Method

Mass and momentum balances give the equation of continuity and the equations of motion, respectively with the rectangular coordinates (x, y) for two-dimensional incompressible Newtonian fluid flow [4], as follows.

$$(\nabla \cdot V)=\partial V_x/\partial x+\partial V_y/\partial y=0, \tag{1}$$

$$\rho\partial V_x/\partial t=-\rho\partial V_x^2/\partial x-\rho\partial(V_xV_y)/\partial y -\partial\tau_{xx}/\partial x-\partial\tau_{yx}/\partial y-\partial p/\partial x, \tag{2}$$

$$\rho\partial V_y/\partial t=-\rho\partial(V_xV_y)/\partial x-\rho\partial V_y^2/\partial y -\partial\tau_{xy}/\partial x-\partial\tau_{yy}/\partial y-\partial p/\partial y, \tag{3}$$

where V_x, V_y are components of the flow velocity, ρ is the density of the fluid. The shear stresses τ_{xx}, τ_{xy}, τ_{yx} and τ_{yy} are defined by

$$\tau_{xx}=-2\mu\partial V_x/\partial x+2\mu(\nabla\cdot V)/3, \tag{4}$$

$$\tau_{xy}=\tau_{yx}=-\mu(\partial V_x/\partial y+\partial V_y/\partial x), \tag{5}$$

$$\tau_{yy}=-2\mu\partial V_y/\partial y+2\mu(\nabla\cdot V)/3, \tag{6}$$

with the viscosity μ. Put $\nu=\mu/\rho$. From Eqs. (1) to (6), one gets

$$\partial V_x/\partial t=-\partial V_x^2/\partial x-\partial(V_xV_y)/\partial y+2\nu\partial^2V_x/\partial x^2 +\nu\left(\frac{\partial^2V_x}{\partial y^2}+\frac{\partial^2V_y}{\partial x\partial y}\right)-\frac{1}{\rho}\partial p/\partial x, \tag{7}$$

$$\partial V_y/\partial t=-\partial(V_xV_y)/\partial x-\partial V_y^2/\partial y+2\nu\partial^2V_y/\partial y^2 +\nu\left(\frac{\partial^2V_y}{\partial x^2}+\frac{\partial^2V_x}{\partial y\partial x}\right)-\frac{1}{\rho}\partial p/\partial y. \tag{8}$$

The expression of Eqs. (7) and (8) contains all velocity components inside the partial derivative sign ∂. The conventional Navier-Stokes equations in the text book [4] have some velocity components outside the derivative sign ∂, by manipulating the convective terms with the help of the equation of continuity (1).

Before finite differentiation is applied, x, y and t in Eqs. (7) and (8) are transformed into dimensionless variables, $\xi=x/L$, $\eta=y/L$ and $\theta=V_b\cdot t/L$, where L is the maximum width of a flow channel in question, V_b is the velocity in the direction of the bulk flow. Reynolds number is defined as $Re=V_b\cdot L/\nu$, and the particle Reynolds number is defined as $Re_p=ReR/L$ with the radius R of a circular obstacle. K is defined as $1/(\rho\cdot L)$.

Corresponding to Eqs. (7) to (8), one obtains

$$\partial U_x/\partial\theta=-\partial U_x^2/\partial\xi-\partial(U_xU_y)/\partial\eta +2\partial^2U_x/\partial\xi^2/Re+\left(\frac{\partial^2U_x}{\partial\eta^2}+\frac{\partial^2U_y}{\partial\xi\partial\eta}\right)/Re -K\partial p/\partial\xi, \tag{9}$$

$$\partial U_y/\partial\theta=-\partial(U_xU_y)/\partial\xi-\partial U_y^2/\partial\eta +2\partial^2U_y/\partial\eta^2/Re+\left(\frac{\partial^2U_y}{\partial\xi^2}+\frac{\partial^2U_x}{\partial\eta\partial\xi}\right)/Re -K\partial p/\partial\eta, \tag{10}$$

These equations are written in the conservative form with dimensionless velocities $U_x=V_x/V_b$ and $U_y=V_y/V_b$.

The domain of interest is discretized by using square mesh with the mesh spacing $\Delta\xi=\Delta\eta=1/N$, where N is the number of subdivisions of the lenght L in the direction of both ξ and η. The mesh points are denoted by (n, m) with $n, m=1, 2, 3, \cdots, N$.

Equations (9) and (10) are approximated by finite differences with respect to the space variables ξ and η. The convective terms are approximated by central finite differences as follows,

$$\partial U_x^2/\partial\xi=(U_{x,n+1,m}^2-U_{x,n-1,m}^2)/(2\Delta\xi), \tag{11}$$

$$\partial U_xU_y/\partial\eta=(U_{x,n,m+1}\cdot U_{y,n,m+1}-U_{x,n,m-1}\cdot U_{y,n,m-1})/(2\Delta\eta), \tag{12}$$

$$\partial U_y^2/\partial\eta=(U_{y,n,m+1}^2-U_{y,n,m-1}^2)/(2\Delta\eta), \tag{13}$$

$$\partial U_xU_y/\partial\xi=(U_{x,n+1,m}\cdot U_{y,n+1,m}-U_{x,n-1,m}\cdot U_{y,n-1,m})/(2\Delta\xi). \tag{14}$$

Momentum transport terms are approximated, respectively, as follows,

$$\partial^2U_x/\partial\xi^2=(U_{x,n+1,m}-2U_{x,n,m}+U_{x,n-1,m})/\Delta\xi^2, \tag{15}$$

$$\partial^2U_x/\partial\eta^2=(U_{x,n,m+1}-2U_{x,n,m}+U_{x,n,m-1})/\Delta\eta^2, \tag{16}$$

$$\partial^2U_y/\partial\xi\partial\eta=\{(U_{y,n,m}-U_{y,n,m-1})-(U_{y,n-1,m}-U_{y,n-1,m-1})\}/(\Delta\xi\Delta\eta), \tag{17}$$

$$\partial^2U_y/\partial\eta^2=(U_{y,n,m+1}-2U_{y,n,m}+U_{y,n,m-1})/\Delta\eta^2, \tag{18}$$

$$\partial^2U_y/\partial\xi^2=(U_{y,n+1,m}-2U_{y,n,m}+U_{y,n-1,m})/\Delta\xi^2, \tag{19}$$

$$\partial^2U_x/\partial\eta\partial\xi=\{(U_{x,n,m}-U_{x,n-1,m})-(U_{x,n,m-1}-U_{x,n-1,m-1})\}/(\Delta\eta\Delta\xi). \tag{20}$$

Attention is paid for Eqs. (17) and (20), where the mesh node for $n-1$ and $m-1$ is utilized. The momentum transport terms give the asymmetrically finite difference model with respect to mesh node (n, m). The conventional Navier-Stokes equation has never referred to this mesh node $(n-1, m-1)$ and gives the symmetrical finite differenced model.

The pressure gradient between adjacent mesh nodes is assumed to be proportional to the difference quatient in kinetic energy. Central finite difference formula is applied to yield

$$\partial p/\partial\xi=k\{(U_{x,n+1,m}^2+U_{y,n+1,m}^2)-(U_{x,n-1,m}^2+U_{y,n-1,m}^2)\}/(2\Delta\xi), \tag{21}$$

$$\partial p/\partial\eta=k\{(U_{y,n,m+1}^2+U_{x,n,m+1}^2)-(U_{y,n,m-1}^2+U_{x,n,m-1}^2)\}/(2\Delta\eta), \tag{22}$$

where k is a proportional constant.

After difference formulae Eq. (11)-Eq. (20) are substituted into the right hand sides of Eqs. (9) and (10), we obtain a set of ordinary differential equations in θ, which is numerically integrated by the Euler method:

$$U_x(\theta+\Delta\theta)=U_x(\theta)+\Delta\theta\cdot\{F_{cx}(\theta)\cdot N/2 + F_{dx}(\theta)\cdot N^2/Re-K\cdot\partial p/\partial\xi\}, \quad (23)$$

$$U_y(\theta+\Delta\theta)=U_y(\theta)+\Delta\theta\cdot\{F_{cy}(\theta)\cdot N/2 + F_{dy}(\theta)\cdot N^2/Re-K\cdot\partial p/\partial\eta\}, \quad (24)$$

for the square mesh with $\Delta\eta=\Delta\xi=1/N$. Here we denote by the functions $F(\theta)$ the terms obtained from Eqs. (11)-Eq. (20). The pressure gradient terms in Eqs. (23) and (24) are also replaced by the finite differences of Eqs. (21) and (22).

These formulae are derived for two-dimensional unsteady state with incompressible Newtonian fluid, and extension to the other cases is very easy.

3. Programming Examples

Source programs written in a subset of BASIC language are shown in List 1 and List 2. Sources are compiled to 32 bit CPU machine codes by the author's compiler system, named SUPER-NOVA387. NOVA implies Numeric Operations on Vast memory Area. The 32 bit floating-point arithmetics are executed by coprocessor of i80387 NDP.

Grammar of the compiler expects lower case character names from a to z. to be 32 bit integer variable storages. 32 bit CPU i80386 registers direct drive arithmetics can be executed at 5 MIPS. The upper case character names from A to Z, AA to ZZ, Aa to Zz and A0 to Z9 imply 32 bit real variables. The floating arithmetics are processed by coprocessor for these real variables at 0.5 MFLOPS.

Dimension area of the compiler system is to access the protect memory space at the 32 bit addressing mode of i80386 CPU. Gigantic main memory space is directly accessed by the compiler system. With intelligent parser, generated codes by the compiler system can cope with the objects written by assembler programming.

200×400 pixel dots directly correspond to the mesh nodes. No. 405 and 406 memories are work spaces to advance time steps. Kinetic energy at a mesh node is colored at 7 colors modulus of 20 cycles or 140 levels. If the kinetic energy exceeds 2.0, velocities are reduced to zero, to avoid numerical divergence greater than chaos.

The coefficients C and D are defined as $\Delta\theta\cdot N^2/Re$ and $\Delta\theta\cdot N/2$. $R=30$ $[m]$ and $L=400$ $[m]$ are cited in the dynamic simulation with $N=200$ divisions. K_p in List 1 and List 2 is equal to k/ρ.

4. Examples

4.1 Chaos in Wakes of Cylindrical Obstacle

Kinetic energy in the wake behind a circular cylinder is obtained by List 1. For $Re=200$, the graphic patterns are shown in Fig. 1 at $K_p=0.5$, with $\Delta\theta=0.000625$. With slight increase, $\Delta\theta=0.00063$, bifurcations to chaos are seen outside the wake as a ghost image, as shown in Fig. 2. At $Re=200$, the coefficient $C=\Delta\theta\cdot N^2/Re$ greater than 0.124 produces numerical instability.

Patterns at $Re=400$ with $\Delta\theta=0.00126$ are shown in Fig. 3. At $Re=400$, the coefficient C greater than 0.125 induces the chaos. Instability is observed both in the outer edge and in the wake similar to the cancer cell growth. The chaotic pattern prevails as time proceeds owing to the propagation of numerical integration errors.

At $Re=800$, chaos proceed only inside the wake as shown in Fig. 4. At $Re=800$, the finite time steps $0.00063\leqq\Delta\theta\leqq0.00074$ give the bifurcation pattern inside the wake, while with $\Delta\theta\geqq0.00075$, chaotic random patterns grow. Bifurcation causes halftone colors in Figs. 2 to 4 until chaotic pattern grows. This instability is not the true image of fluid flow, but the result of the nonlinear discrete dynamics. Chaos comes from the numerical instability of time integration applied to the nonlinear system [1-3]. The momentum transport term is known to be effective in stabilizing the numerical solution. The coefficient $D=\Delta\theta\cdot N/2$ in the convective terms is less than 0.1.

4.2 Chaos Trailing on Joukowski Aerofoil

A Joukowski aerofoil with the angle of attack 18° is used as another numerical example to test the numerical method. In Figs. 5 and 6, the kinetic energy around the aerofoil is illustrated for $\Delta\theta=0.00125$ and 0.00126, respectively. With $Re=400$, $K_p=0.5$, the pattern shows the same stability properties as in the case of a circular cylindrical obstacle.

5. Conclusion

The numerical method developed in this work is useful to check the numerical instability. Chaos is formed by the numerical instability caused by a large time step after the bifurcation patterns. The threshold of the time step for the stable integration is determined by numerical experiments for the present method.

The algorithm of the method is fitted for papallel processor or array processor. Extension to the three-dimensional fluid flow is easy with the new method.

Moreover, the transient behavior of the energy pattern should be visualized besides flow pattern visualization, to check the propagation of the numerical integration errors [2, 3, 5].

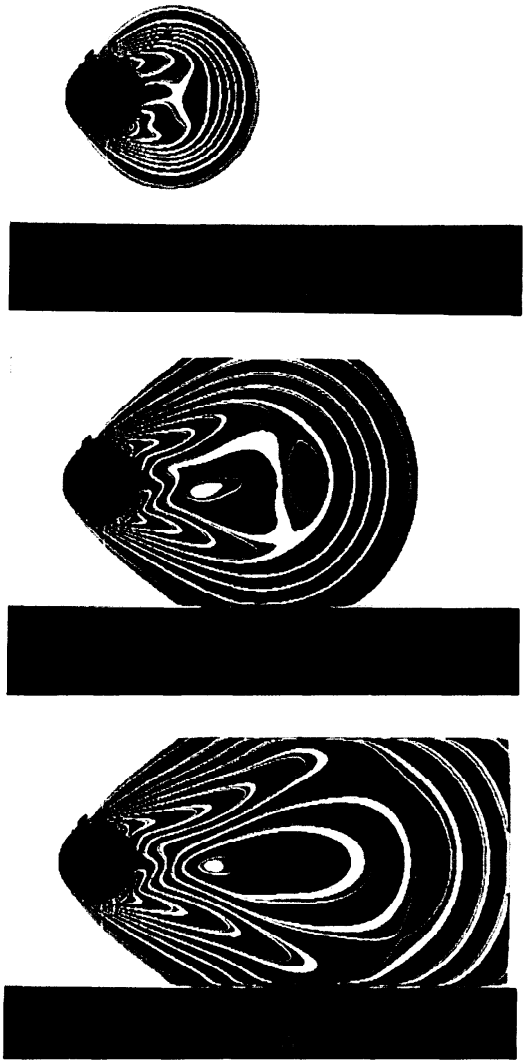

Fig. 1 Kinetic Energy, $Re=200$, $\Delta\theta=0.000625$, Circular Obstacle.

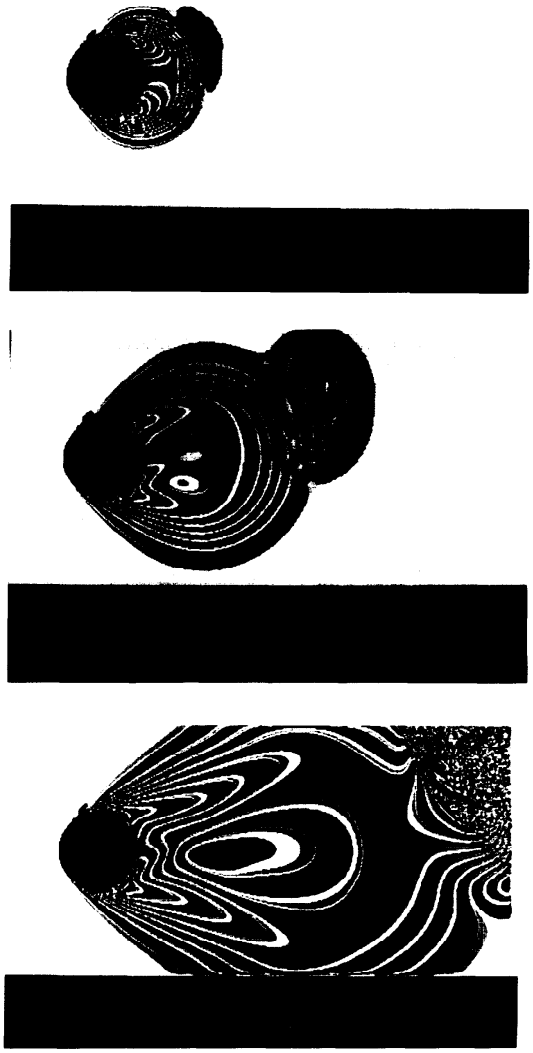

Fig. 2 Kinetic Energy, $Re=200$, $\Delta\theta=0.00063$, Circular Obstacle. Chaos Generation.

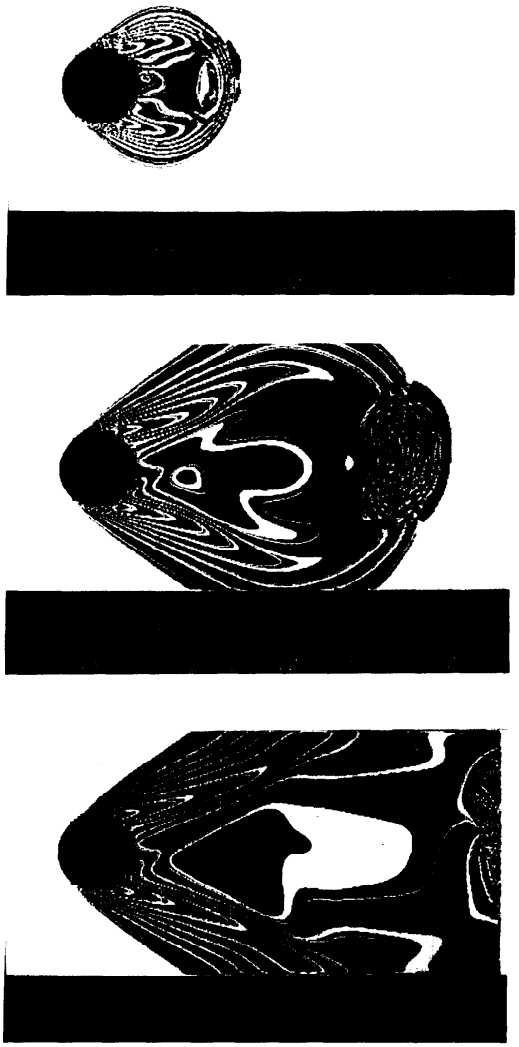

Fig. 3 Kinetic Energy, $Re=400$, $\Delta\theta=0.00126$, Circular Obstacle. Chaos Generation.

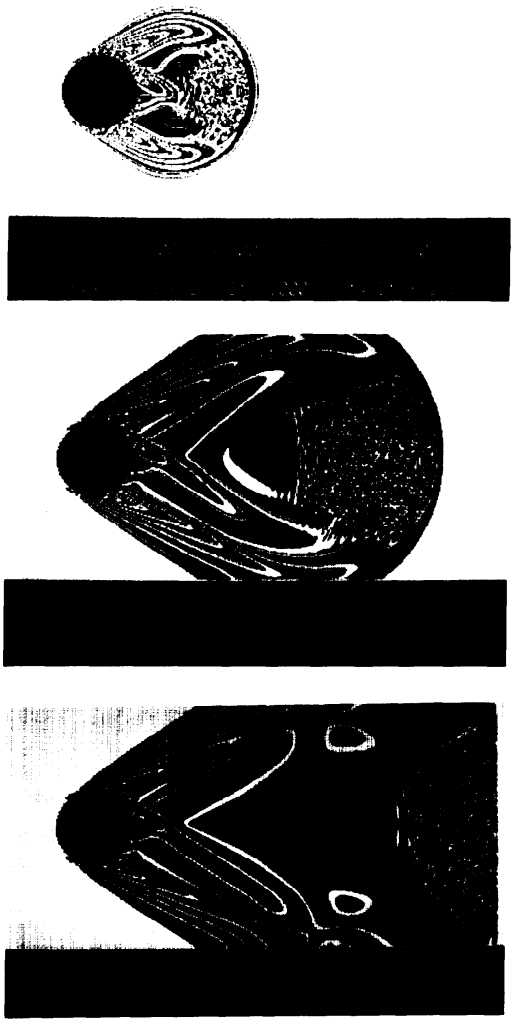

Fig. 4 Kinetic Energy, $Re=800$, $\Delta\theta=0.00075$, Circular Obstacle. Chaos Generation.

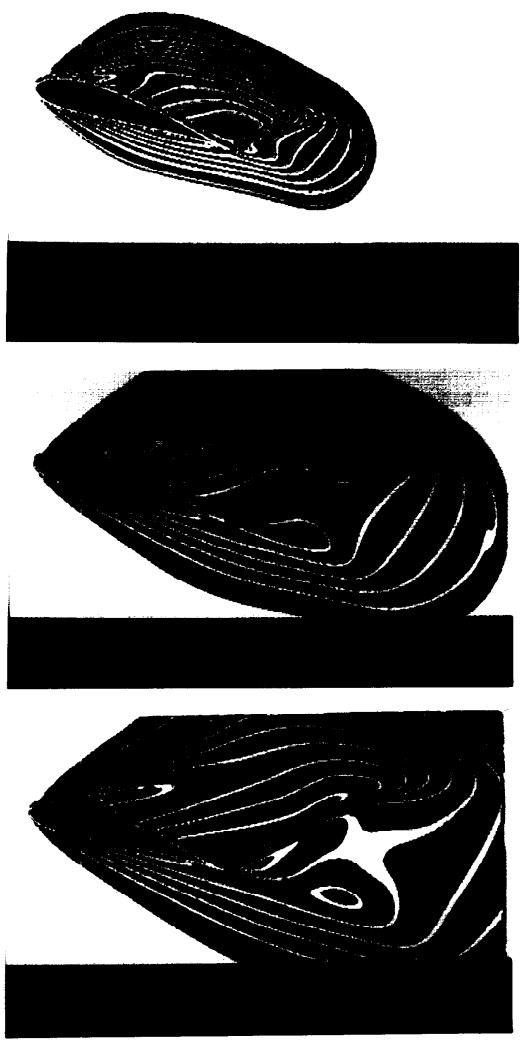

Fig. 5 Kinetic Energy, $Re=400$, $\Delta\theta=0.00125$, Joukowski Aerofoil.

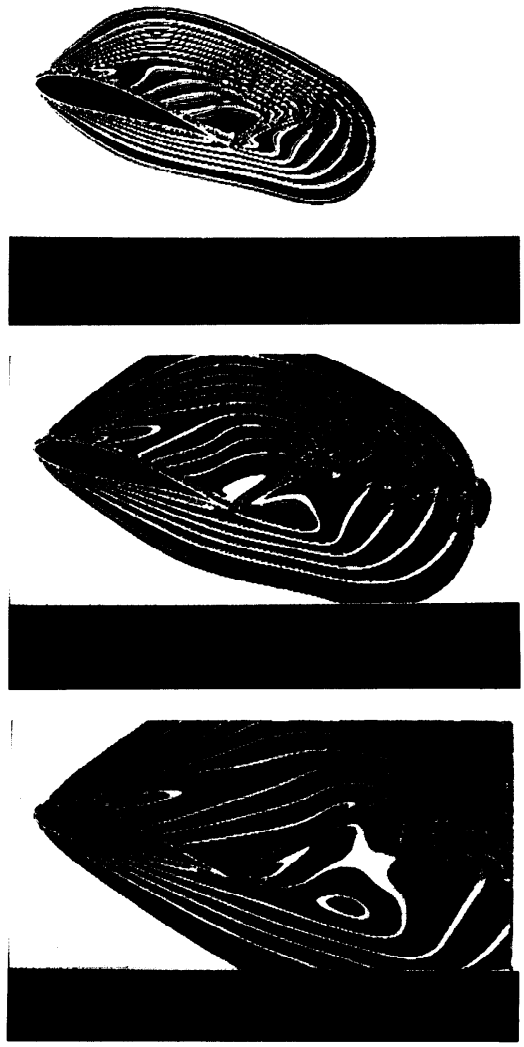

Fig. 6 Kinetic Energy, $Re=400$, $\Delta\theta=0.00126$, Joukowski Aerofoil. Chaos Generation.

List 1 Source Program for Circular Obstacle.

```
 1]"CIR426"2D Fluid Dynamics (Circular Object) by T.Aratani Oct. '90
 2]DIM U(406,200),V(406,200)    :CLS       :CLS 2  :Kp=0.5          :Lm=2
 3] Dt=0.00126:n=200:N=CSNG(n):Re=400    :U0=1    :R=30:R2=R*R    :m=n-1
 4] DN=Dt*N    :C=N*DN/Re       :D=DN*0.5          :r=CINT(R2)   :s=n*2-1
 5] LINE(0,0)-(639,399),1,BF:LINE (100,101)-(400,299),0,BF
 6]       LOCATE 15,21:? "Circle   Re=";Re;"  N=";N;"  C=";C;"  D=";D
 7]     FOR i=0 TO 406
 8]          FOR j=0 TO n                  :U(i,j)=U0:V(i,j)=0
 9]       x=i-75 :y=j-100:z=x*x:z=y*y+z:IF z<r THEN U(i,j)=0:GOTO 10
10]         U=U(i,j):V=V(i,j):UV=U*U+(V*V)
11]       p=CINT(UV*139):p=p MOD 7+1:a=i+100:b=-j+300:PSET(a,b),p
12]       10 NEXT j:ON STOP
13]     NEXT i:       LOCATE 20,23:? "Kp=";Kp;"  Lm=";Lm
14]FOR t=0 TO 350: LOCATE 36,23:? "Dt=";Dt;"  t= Dt *";t
15]     FOR i=1 TO s
16]          FOR j=1 TO m:x=i-75 :y=j-100 :z=x*x:z=y*y+z:IF z<r THEN 30
17]     X=U(i,j):P=U(i,j+1):Q=U(i,j-1):E=U(i+1,j):F=U(i-1,j):O=U(i-1,j-1)
18]     Y=V(i,j):U=V(i,j+1):V=V(i,j-1):G=V(i+1,j):H=V(i-1,j):K=V(i-1,j-1)
19]  FF=F*F:EE=E*E:FH=H*H+FF:EG=G*G+EE:UU=U*U:VV=V*V:PU=P*P+UU:QV=Q*Q+VV
20]             Sx=Kp*(FH-EG)   : Sy=Kp*(QV-PU)
21]      X2=X+X:    Z=E+F-X2     :Z=Z+Z+P+Q-X2    :Z=Z+Y+K-V-H   :W=FF-EE
22]                 S=Q*V-(P*U)  :W=W+S+Sx:S=Z*C  :A=W*D+S+X     :U(405,j)=A
23]      Y2=Y+Y:    Z=U+V-Y2     :Z=Z+Z+G+H-Y2    :Z=Z+X+O-Q-F   :W=VV-UU
24]                 S=F*H-(E*G)  :W=W+S+Sy:S=Z*C  :B=W*D+S+Y     :V(405,j)=B
25]                 AB=A*A+(B*B):IF AB>Lm THEN U(405,j)=0:V(405,j)=0
26]               p=CINT(AB*139):p=p MOD 7+1:a=i+100:b=-j+300:PSET(a,b),p
27]      30   NEXT j:ON STOP
28]          FOR j=1 TO m:   U(0,j)=U0:V(0,j)=0
29]                        x=i-75 :y=j-100 :z=x*x:z=y*y+z:IF z<r THEN 40
30]                        U(i-1,j)=U(406,j):U(406,j)=U(405,j)
31]                        V(i-1,j)=V(406,j):V(406,j)=V(405,j)
32]      40   NEXT j:ON STOP
33]     NEXT i:ON STOP
34]NEXT t:STOP
```

List 2 Source Program for Joukowski Aerofoil.

```
 1]"ATC426"2D Fluid Dynamics (Aerofoil) by T.Aratani, Nov. '90
 2]DIM U(406,200),V(406,200),W(406,200)    :CLS     :CLS 2 :Kp=0.5:Lm=2
 3] Dt=0.00126:n=200:N=CSNG(n)   :Re=400    :U0=1    :R=30:R2=R*R :m=n-1
 4]     LINE(10,1)-(610,398),1,BF:LINE(101,110)-(480,300),0,BF
 5] DN=Dt*N   :C=N*DN/Re          :D=DN*0.5          :r=CINT(R2) :s=n*2-1
 6]LOCATE 14,21:? "Joukowski 18°  Re=";Re;"  N=";N;"  C=";C;"  D=";D
 7]             Pi=3.141593:RT=-Pi/10:CS=COS(RT):SN=SIN(RT)
 8]FOR i=0 TO 406:FOR j=0 TO n:W(i,j)=0:NEXT j:NEXT i :P=-5:Q=3:RC=45
 9]    RR=RC*RC-(Q*Q):SR=SQR(RR):Pa=ABS(P):A=-Pa/SR+1:A=RC*A
10]    A2=A*A:Pp=Pi/5000:FOR i=0 TO 10001:I=CSNG(i):IP=I*Pp
11]    Si=SIN(IP):Co=COS(IP):R=RC*Co+P:I=RC*Si+Q:RI=R*R+(I*I)
12]    AR=A2/RI:X=R*(1+AR):Y=I*(1-AR):XX=X*CS-(Y*SN):YY=X*SN+(Y*CS)
13]    x=CINT(XX):y=CINT(YY):p=x+200:q=200-y
14]    PSET(p,q),7:x=x+100:y=y+100:W(x,y)=1:NEXT i
15]     FOR i=0 TO 406   :f=0
16]          FOR j=0 TO n          :U(i,j)=0:V(i,j)=0
17]          W=W(i,j):   IF W=1 THEN f=1
18]                      IF f=1 THEN 80
19]                      U(i,j)=U0:U=U(i,j):V=V(i,j):UV=U*U+(V*V)
20]       p=CINT(UV*34):p=p MOD 7+1:a=i+100:b=-j+300:PSET(a,b),p
21]       80 NEXT j:ON STOP
22]     NEXT i
23]     FOR i=0 TO 406   :f=0
24]          FOR j=0 TO n :k=n-j
25]          W=W(i,k):   IF W=1 THEN f=1
26]                      IF f=1 THEN 90
27]                      U(i,k)=U0:U=U(i,k):V=V(i,k):UV=U*U+(V*V)
28]       p=CINT(UV*139):p=p MOD 7+1:a=i+100:b=-k+300:PSET(a,b),p
29]       90 NEXT j:ON STOP
30]     NEXT i       :LOCATE 20,23:? " Kp=";Kp;"  Lm=";Lm
31]     FOR i=0 TO 406:FOR j=0 TO n:W(i,j)=U(i,j):NEXT j:NEXT i
```

List 2 Source Program for Joukowski Aerofoil.

```
32]FOR t=0 TO 1320 :LOCATE 36,23:? " Dt=";Dt;"  t= Dt *";t
33]     FOR i=1 TO s
34]           FOR j=1 TO m:W=W(i, j):IF W=0 THEN 30
35] X=U(i, j):P=U(i, j+1):Q=U(i, j-1):E=U(i+1, j):F=U(i-1, j):O=U(i-1, j-1)
36] Y=V(i, j):U=V(i, j+1):V=V(i, j-1):G=V(i+1, j):H=V(i-1, j):K=V(i-1, j-1)
37]                FF=F*F:EE=E*E:FH=H*H+FF:EG=G*G+EE
38]                UU=U*U:VV=V*V:PU=P*P+UU:QV=Q*Q+VV
39]                Sx=Kp*(FH-EG):Sy=Kp*(QV-PU)
40]      X2=X+X:    Z=E+F-X2      :Z=Z+Z+P+Q-X2    :Z=Z+Y+K-V-H   :W=FF-EE
41]                 S=Q*V-(P*U)  :W=W+S+Sx:S=Z*C  :A=W*D+S+X   :U(405, j)=A
42]      Y2=Y+Y:    Z=U+V-Y2      :Z=Z+Z+G+H-Y2    :Z=Z+X+O-Q-F   :W=VV-UU
43]                 S=F*H-(E*G)  :W=W+S+Sy:S=Z*C  :B=W*D+S+Y   :V(405, j)=B
44]                 AB=A*A+(B*B):IF AB>Lm THEN U(405, j)=0:V(405, j)=0
45]            p=CINT(AB*139):p=p MOD 7+1:a=i+100:b=-j+300:PSET(a, b), p
46]      30   NEXT j:ON STOP
47]           FOR j=1 TO m:U(0, j)=U0:V(0, j)=0:W=W(i, j):IF W=0 THEN 40
48]                        U(i-1, j)=U(406, j):U(406, j)=U(405, j)
49]                        V(i-1, j)=V(406, j):V(406, j)=V(405, j)
50]      40   NEXT j:ON STOP
51]     NEXT i:ON STOP
52]NEXT t:STOP
```

References

1. ARATANI, T. Fractals from Nonlinear Ordinary Differential Equations (in Japanese), *Trans. IPS Japan*, **29** (1988), 1-6.
2. ARATANI, T. Generation of Chaos and Fractals from Nonlinear Control System (in Japanese), *Journal of The Society of System, Control and Information*, **34** (1990), 98-105.
3. ARATANI, T. Computer Graphic Display Methodology for Management System Stability, *Bulletin of Hiroshima Prefectural University*, **1** (1990), 81-93.
4. BIRD, R. B., STEWART, W. E. and RIGHTFOOT, E. N. Transport Phenomena, Wiley International Edition (1960), 74-81.
5. DAIGUUJI, H. Finite Difference Analysis for Navier-Stokes Equations (in Japanese), *Math. Sci.* (Suuri Kagaku), 263 (1985), 38-45.
6. HASHIMOTO, H. Characteristics and Structure of Incompressible Fluid (in Japanese), *Math. Sci.* (Suuri Kagaku), 263 (1985), 25-31.
7. ISHII, K. Numerical Simulation of Turbulent Phenomena (in Japanese), *Math. Sci.* (Suuri Kagaku), 314 (1989), 43-56.
8. KUWAHARA, K. Numerical Simulation of Compressible Navier-Stokes Equations (in Japanese), *Math. Sci.* (Suuri Kagaku), 263 (1985), 32-37.
9. MIZUKAMI, K. Computer Graphics—Informative Society and Images— (in Japanese), Asakura Books (1989).
10. MIZUNO, A. Introduction to Numerical Flow Analysis (in Japanese), Asakura Books Co. (1990).
11. TATSUMI, T. General Remarks around the Navier-Stokes Equations (in Japanese), *Math. Sci.* (Suuri Kagaku), 263 (1985), 5-15.

(Received December 25, 1990; revised June 25, 1991)