Numerical Study of Traveling Wave Solutions to 2D Boussinesq Equation

Posing the problem

- The accurate derivation[1] of the Boussinesq system combined with an approximation, that reduces the full model to a single equation, leads to the Boussinesq Paradigm Equation[2] (BPE):

$$u_{tt} - \Delta u - \beta_1 \Delta u_{tt} + \beta_2 \Delta^2 u + \Delta F(u) = 0, \qquad F(u) := \alpha u^2 .$$

- The equation admits 2D stationary translating solition solution. Unfortunately the propagating soliton in 2D is very 'fragile' and transforms into smaller vawes as time progress.

[1]Christov, C. I. 2001, ‘An energy-consistent Galilean-invariant dispersive shallow-water model’, Wave Motion 34, 161–174.
[2]Christov, C. I. 1995a, Conservative difference scheme for Boussinesq model of surface waves, in, In: ‘Proc. ICFD V’, Oxford University Press, pp. 343–349.
[3]C. I. Christov, J. Choudhury, Perturbation solution for the 2D Boussinesq equation, Mech. Res. Commun., 38 (2011) 274 - 281.

Here, once again, we seek for travelling wave solutions for the BPE in drection *y*, with phase velocity c, i.e. solutions of type u(*x*,*y*,*t*) = U(*x*,*y*-*ct*):

$$c^2(E-\beta_1\Delta)U_{yy} - \Delta U + \beta_2\Delta^2 U + \Delta F(U) = 0. \quad (BE)$$

The variable change

$$x = \sqrt{\beta_1}\,\bar{x}, \quad y = \sqrt{\beta_1}\,\bar{y}, \quad v(\bar{x},\bar{y}) := U(x,y)$$

transforms (BE) to:

$$\beta c^2(E-\Delta)v_{\bar{y}\bar{y}} - \beta\Delta v + \Delta^2 v + \alpha\beta\Delta(v^2) = 0, \quad (vcBE)$$

with $\beta = \beta_1 / \beta_2$ and $\alpha, \beta > 0$. Equation (vcBE) can be rewritten as a system of two elliptic equations of second order:

$$-(1-c^2\beta)\Delta v + \beta(1-c^2)v + \alpha\beta\theta v^2 = w$$

$$-\Delta w = c^2\beta(E-\Delta)v_{xx}$$

Algorithm

The trivial solution is always present and must be avoided. Thus we proceed as in [4] and fix the value of the function in one point in order to prevent the iterative algorithm of ‘slipping’ into the trivial solution. For definiteness we take $v(0,0) = \theta$ and then introduce $\tilde{v} = \theta v$ and $\hat{w} = \theta w$:

$$-(1-c^2\beta)\Delta\hat{v} + \beta(1-c^2)\hat{v} + \alpha\beta\theta\hat{v}^2 = \hat{w} \qquad (SYS.1)$$

$$-\Delta\hat{w} = c^2\beta(E-\Delta)\hat{v}_{xx} \qquad (SYS.2)$$

The value of θ is found from the equation

$$\theta = \left.\frac{(1-c^2\beta)\Delta\hat{v} - \beta(1-c^2)\hat{v} + \hat{w}}{\alpha\beta}\right|_{x=0,\,y=0} \qquad (TH)$$

[4]C. I. Christov, Numerical implementation of the asymptotic boundary conditions for steadily propagating 2D solitons of Boussinesq type equation, Math. Computers Simul., 82 (2012) 1079 - 1092.

Since, we step into new problem, with further considerations on the new boundary condition, we chose the simple iteration method[4] (false transients) as one of the simplest possible approaches. Thus we introduce artifical time, and add false time derivatives to get

$$\frac{\partial \hat{v}}{\partial t} - (1-c^2\beta)\Delta\hat{v} + \beta(1-c^2)\hat{v} + \alpha\beta\theta\hat{v}^2 = \hat{w} \quad (tdSYS.1)$$

$$\frac{\partial \hat{w}}{\partial t} - \Delta\hat{w} = c^2\beta(E-\Delta)\hat{v}_{xx}. \quad (tdSYS.2)$$

Now the solution to the steady coupled elliptic system (SYS) is replaced by solving the pertinent transient equations (tdSYS) until their solutions $\tilde{v}$ and $\hat{w}$ cease to change signifcantly in time.

Milestones:

- Use high order finite differences which allow higher accuracy on relatively coarser grids.

- Update the existing asymptotics of the equation and implement the new boundary condition

- Find algorithm's convergence rate using nested grids

- Optimize the algorithm
  - Resolve proper time step for each time iteration
  - Create a good implementation of the discrete Laplacian operator as more than 50% of the algorithm running time is spent on its calculation

Discretization

For simplicity we use a uniform grid with equal step size h along both x,y-axes directions. The value of the function $\tilde{v}$ at mesh point (x_i, y_j, t_k) is denoted by:

$$\hat{v}_{i,j}^{k} := \hat{v}(x_i, y_j, t_k)$$

We discretize the spatial derivatives in (tdSYS) using centered finite differences and extending the stencil:

$$v_{\hat{x}\hat{x},p}(x) := \frac{1}{h^2} \sum_{i=-p/2}^{p/2} d_i v(x+ih)$$

Thus:

$$\Delta_{h,p} v_{i,j} := (v_{i,j})_{\hat{x}\hat{x},p} + (v_{i,j})_{\hat{y}\hat{y},p}$$

Here p defines the order of the finite difference. The weights d_i are defined as follows:

$$h^2 \frac{\partial^2}{\partial \hat{x}^2} = h^2 \frac{\partial^2}{\partial \hat{y}^2}$$

p = 2: $$[1 \quad -2 \quad 1]$$

p = 4: $$\left[-\frac{1}{12}, \frac{4}{3}, -\frac{5}{2}, \frac{4}{3}, -\frac{1}{12}\right]$$

p = 6: $$\left[\frac{1}{90}, -\frac{3}{20}, \frac{3}{2}, -\frac{49}{18}, \frac{3}{2}, -\frac{3}{20}, \frac{1}{90}\right]$$

The solutions at the t^{k+1} time level are evaluated directly:

$$\frac{\hat{v}_{i,j}^{k+1} - \hat{v}_{i,j}^{k}}{\tau} - (1 - c^2\beta)\Delta_{h,p}\hat{v}_{i,j}^{k} + \beta(1 - c^2)\hat{v}_{i,j}^{k} + \alpha\beta\theta\left(\hat{v}_{i,j}^{k}\right)^2 = \hat{w}_{i,j}^{k} \qquad (FD.1)$$

$$\frac{\hat{w}_{i,j}^{k+1} - \hat{w}_{i,j}^{k}}{\tau} - \Delta_{h,p}\hat{w}_{i,j}^{k} = c^2\beta(E_h - \Delta_{h,p})\left(\hat{v}_{i,j}^{k}\right)_{\hat{x}\hat{x}}. \qquad (FD.2)$$

Asymptotic conditions on the boundaries

The behavior of the solutions $\tilde{v}$ and $\hat{w}$ as r = sqrt(x^2+y^2) goes to infinity is studied in details in [3, 4]. From the numerical results given there it follows that $\tilde{v}$ and $\hat{w}$ have O(r^{-2}) asymptotic decay at infinity. We suppose that the second order derivatives of $\tilde{v}$ and $\hat{w}$ are of order O(r^{-4}) whereas fourth order derivatives and the nonlinear term in equation (vcBE) are of order O(r^{-6}).

$$\beta c^2 (E-\Delta) v_{yy} - \beta \Delta v + \Delta^2 v + \alpha\beta\Delta(v^2) = 0, \quad (vcBE)$$

As we consider equation (vcBE) for sufficiently large r and insert the asymptotic values of all terms in it and neglect the higher order in r terms (of order O(r^{-6})). In this way we obtain the following problem valid for large r

$$c^2 v_{yy} = \Delta v, \quad \text{(infEQ)}$$

$$v(x,y), \Delta v(x,y) \longrightarrow \quad \text{as} \quad \sqrt{x^2+y^2} \longrightarrow \propto$$

The function that resolves the equation (infEQ) is

$$v(x,y)=\mu\left(\overline{c}x^2+y^2\right)^{-q/2}\cos\left(q\arccos\left(\frac{\overline{c}x}{\sqrt{\overline{c}x^2+y^2}}\right)\right)\qquad (BND)$$

$$\overline{c}=1-c^2$$

with some positive real numbers μ and q. Using the asymptotic decay of the solution and its symmetry with respect to coordinate axes, we obtain q = 2 and thus:

$$\overline{v}(x,y)=\mu\frac{\left(\overline{c}x^2-y^2\right)}{\left(\overline{c}x^2+y^2\right)^2}\qquad (vB)$$

$$\overline{w}(x,y)=\overline{\mu}\frac{\left(\overline{c}x^2-y^2\right)}{\left(\overline{c}x^2+y^2\right)^2}.\qquad (wB)$$

In order to resolve the boundary functions (vB) and (wB) completely one needs the μ parameters. We obtain them iteratively, at each time level of (), by the following minimizatoin procedure. For given solutions $\tilde{v}_{i,j}{}^{k}$ and $\hat{w}_{i,j}{}^{k}$ at time level t^k we choose μ and $\bar{\mu}$ as minimizers of:

$$\mu = \min_{\mu > 0} \left\| \bar{v}(x_i, y_i) - \hat{v}_{i,j}^k) \right\|_{L2,\bar{\Omega}}$$

$$\bar{\mu} = \min_{\bar{\mu} > 0} \left\| \overline{w}(x_i, y_i) - \hat{w}_{i,j}^k) \right\|_{L2,\bar{\Omega}}$$

Tests and simulations showed that $\bar{\Omega}$ should not solely consist of boundary points but should also include some inner points of Ω_h near boundary $\partial\Omega_h$. Each minimization problem produces a simple linear equation, which depends on the type of norm used.

Stop Criteria and time step control

The residual of the discrete approximation to (vcBE) at the mesh point (x_i; y_j ; t_k) is defined as:

$$R^k_{i,j} := \beta c^2 (E_h - \Delta_{h,p})(\hat{v}^k_{i,j})_{\hat{y}\hat{y},p} - \beta \Delta_{h,p} \hat{v}^k_{i,j} + \Delta^2_{h,p} \hat{v}^k_{i,j} + \alpha\beta\theta\Delta_{h,p} \left(\hat{v}^k_{i,j} \right)^2 \quad (R)$$

Our numerical computations show that:

$$\max_{i,j} \left| \hat{v}^{k+1}_{i,j} - \hat{v}^k_{i,j} \right| < \max_{i,j} \left| R^{k+1}_{i,j} \right| < \varepsilon$$

It is possible to use varying time step τ to optimize and speed up the algorithm. When the time step becomes too big the solution starts to diverge and becomes jagged. Fortunately these signs appear first in the residual R in (R) i.e. it starts to grow and jag simultaneously while the solution is still fine. This is a clear sign that the time step has to be decreased, otherwise we can increase it.

Validation - Algorithm's Convergence

$\alpha = 1$

$\beta = 3$

$\Omega_h =$ [0,30]x[0,30]

FDS	h	errors E_i in L_2	Conv. Rate	errors E_i in L_∞	Conv. Rate
c=0.45 $O(h^2)$	0.2				
	0.1	0.037705		0.024736	
	0.05	0.008922	2.0794	0.005810	2.0901
c=0.1 $O(h^2)$	0.2				
	0.1	0.015770		0.015794	
	0.05	0.004854	1.6999	0.005243	1.5908
c=0.45 $O(h^4)$	0.2				
	0.1	0.020795		0.008505	
	0.05	0.000278	6.2269	0.000235	5.1800
c=0.1 $O(h^4)$	0.2				
	0.1	0.000892		0.001463	
	0.05	5.4667e-05	4.0281	9.5747e-05	3.9337
c=0.45 $O(h^6)$	0.4				
	0.2	2.0975e-02		2.9341e-02	
	0.1	3.5348e-04	5.8909	5.8346e-04	5.6521
c=0.1 $O(h^6)$	0.4				
	0.2	3.7059e-03		3.7572e-03	
	0.1	7.4723e-05	5.6321	8.3359e-05	5.4942

Convergence test for (FD) with different approximation errors. Errors E_i are measured in L_2 and L_{inf} norms.

The errors E_i are defined as follows:

$$E_1 = \left\| \hat{v}_{[h]} - \hat{v}_{[h/2]} \right\|, E_2 = \left\| \hat{v}_{[h/2]} - \hat{v}_{[h/4]} \right\|,$$

and the convergence rate (by Runge's formula):

$$(\log E_1 - \log E_2) / \log 2$$

Derivative Convergence

FDS	h	errors in L_2	Conv. Rate	errors in L_∞	Conv. Rate
c=0.45	0.8				
$O(h^2)$	0.4	2.9698e-01		4.2497e-01	
	0.2	6.8742e-02	2.1111	8.6465e-02	2.2972
c=0.1	0.8				
$O(h^2)$	0.4	3.4849e-01		3.0271e-01	
	0.2	8.7696ee-02	1.9905	7.5691e-02	1.9998
c=0.45	0.8				
$O(h^6)$	0.4	1.0766e+00		1.2316e+00	
	0.2	3.5768e-02	4.91117	5.8927e-02	4.3855
c=0.1	0.8				
$O(h^6)$	0.4	8.0095e-01		9.8911e-01	
	0.2	1.5680e-02	5.6747	2.1238e-02	5.5414

Errors in L_2 and L_{inf} norms and convergence rate for fourth order x-derivatve evaluated by the FDS with $O(h^2)$ and $O(h^6)$ approximation order

Runge's test, evaluating the fourth x-derivative of the solution, show that it converges numerically. Tests for other fourth order derivatives are similar and we do not present them here.

Validation of Boundary Condition

$L_x = L_y$	v_{0,N_y}	μ	$\tilde{\mu}$	$\|v - \overline{v}\|_{2,\hat{\Omega}}$	$\|w - \overline{w}\|_{2,\hat{\Omega}}$
20	-2.23e-04	1.9355e-01	1.9353e-01	4.17e-05	9.75e-05
40	-5.65e-05	1.9370e-01	1.9369e-01	4.42e-06	1.03e-05
80	-1.41e-05	1.9378e-01	1.9378e-01	7.56e-07	1.79e-06
160	-3.53e-06	1.9381e-01	1.9381e-01	7.44e-10	1.40e-09

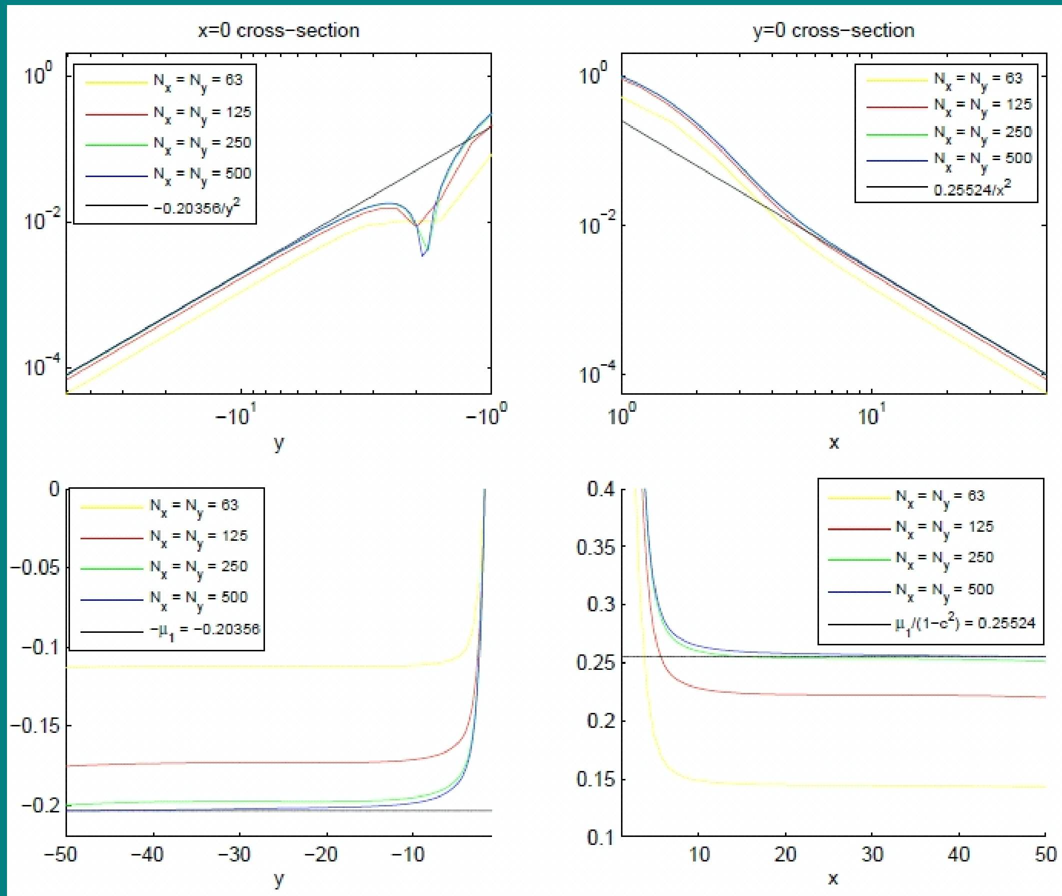
- The boundary values in the 2nd column form a geometric progression with common ratio ¼, which is pure sign of $1/r^2$ asymptotics
- μ parameters settle down as computational box enlarges
- Boundary function becomes more precise for larger domains

x-y cross-sections of the solution

Upper panels:
-The absolute value of the function on log-log plots.
-Black line describes (vB) function with the respective μ parameter

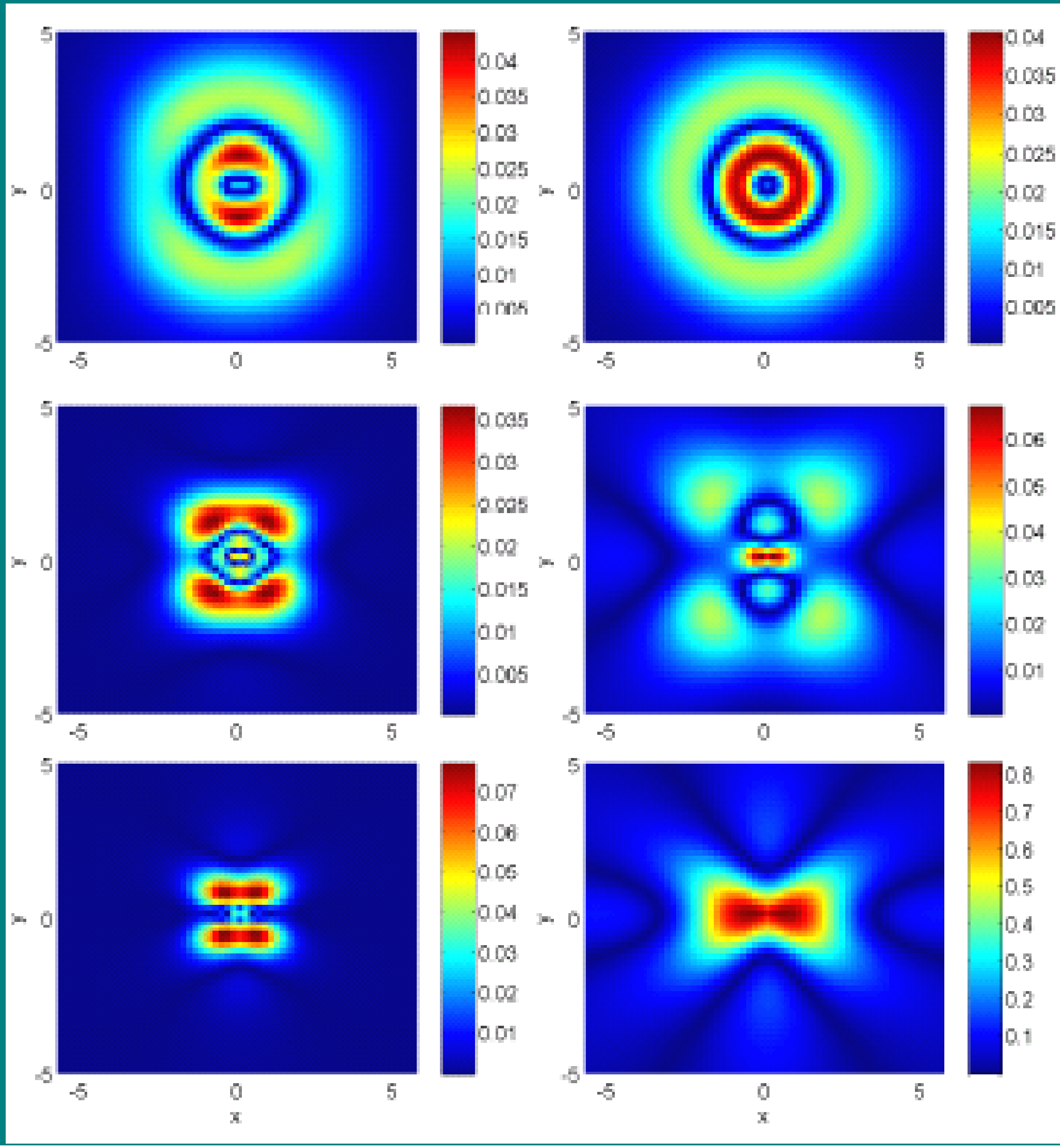
Lower panels show:
-Plots display vr^2 values along the vertical z-axis

The solution settles down as the step size h decreases!



The efect of the mesh size. Lower panels: the function scaled by r^2. N_x,N_y – number of mesh-points along x,y axis.

Difference between the numerical solution $\tilde{v}$ and the best fit formulae [3].



c=0.3
β = 1

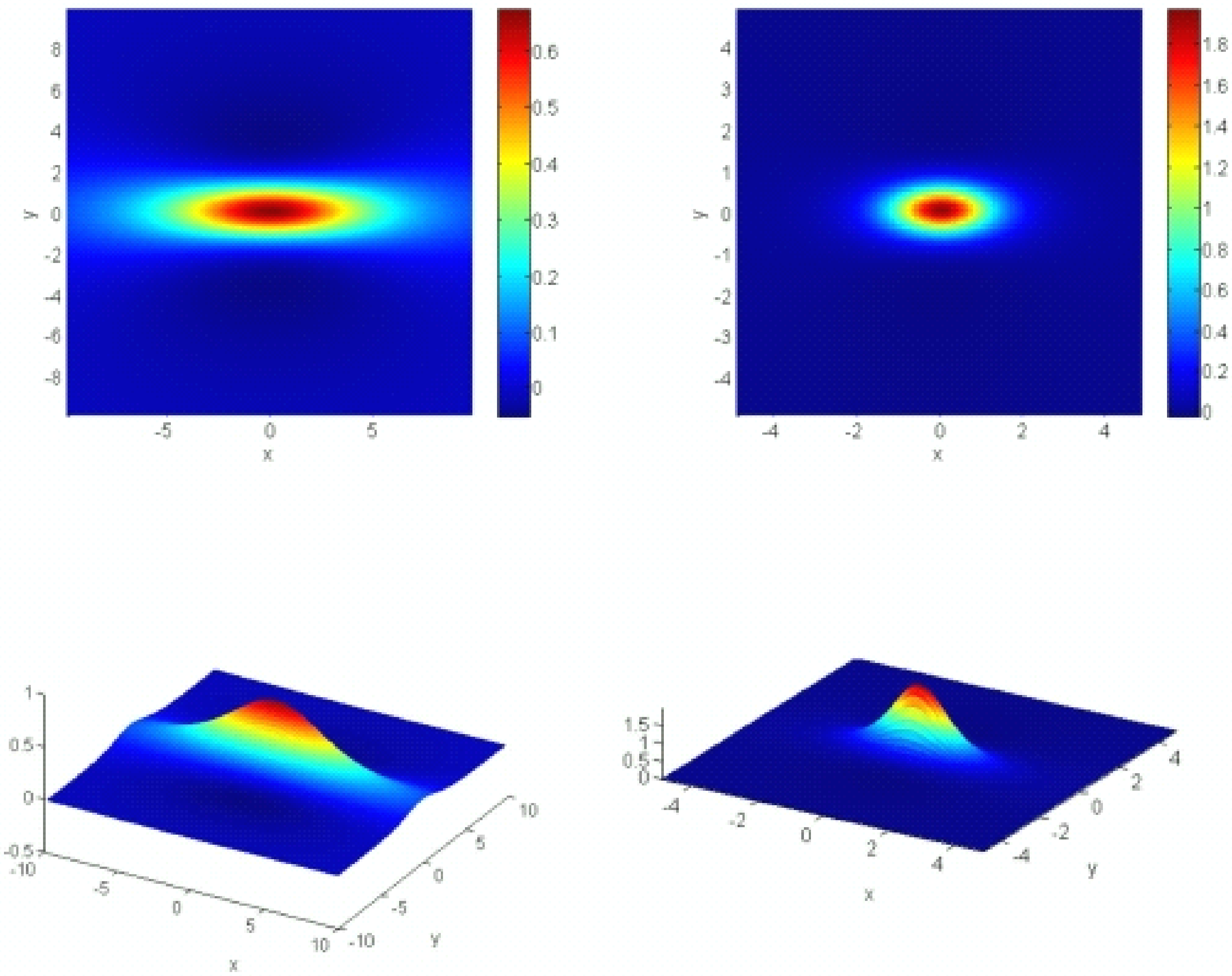
c=0.1
β = 1

c=0.3
β = 3

c=0.5
β = 1

c=0.3
β = 5

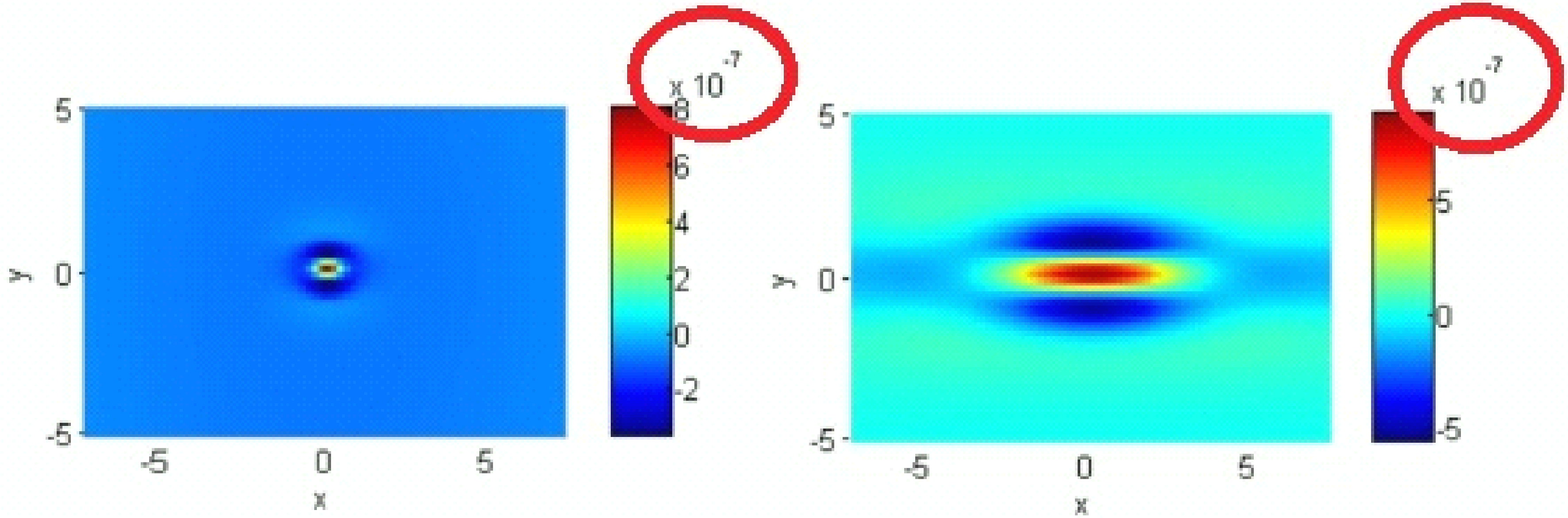
c=0.9
β = 1



$c = 0.9,\ \beta = 1$ $\qquad$ $c = 0.5,\ \beta = 3$

Figure 3: 2D and 3D profiles of the numerical solution

Residual



$\beta = 5,\ c = 0.3$ $\beta = 1,\ c = 0.9$

Residual at the last step of iteration process

Thank you for your attention