

# Coupled PDEs with Initial Solution from Data in COMSOL 4

X. Huang, S. Khuvis, S. Askarian, M. K. Gobbert\*, and B. E. Percy

Department of Mathematics and Statistics, University of Maryland, Baltimore County

\*Corresponding author: 1000 Hilltop Circle, Baltimore, MD 21250, gobbert@umbc.edu

**Abstract:** Many physical applications require the solution of a system of coupled partial differential equations (PDEs). In most cases the analytic PDE solution does not exist for this system and we need to solve the problem numerically using the finite element method in COMSOL. This paper presents information on techniques needed in COMSOL 4 to enable computational studies of coupled systems of PDEs for time-dependent non-linear problems. Furthermore, we demonstrate how to use data files as input for initial conditions. To illustrate the techniques, we consider a system of two time-dependent non-linear PDEs from mathematical biology.

**Keywords:** System of PDEs, Coupled PDEs, Reaction-diffusion equation, Initial condition.

## 1 Introduction

This paper extends the step-by-step instructions in [3, 4] for solving one stationary linear PDE to a system of time-dependent non-linear PDEs. We present two different approaches in COMSOL to solve coupled systems of PDEs. In the first approach, each equation in the coupled system of PDEs is modeled independently in its own **Physics** model, and then they are coupled together. In the second approach, the matrix form of coefficients is used to specify all PDEs of the coupled system in one **Physics** model. This approach is appropriate if all PDEs have the same form. Furthermore, the technique to read initial solution from a data file into COMSOL 4 is introduced.

To illustrate how to set up coupled PDEs with initial solution from a data file, we present an example from mathematical biology, the FitzHugh-Nagumo equations [1]

$$\begin{aligned} C_t - \nabla \cdot (D_{\text{eff}} \nabla C) &= C(C - \alpha)(1 - C) - \beta v, \\ v_t &= \epsilon(C - \gamma v), \end{aligned} \quad (1)$$

inspired by a recent study of calcium dynamics [2]. The spread of excitation variable  $C(x, y, t)$  through

a cell given by the square domain  $\Omega = (0, 150) \times (0, 150) \subset \mathbb{R}^2$  in units of micrometers ( $\mu\text{m}$ ) is described mathematically by this reduced system of coupled time-dependent reaction-diffusion equations for all  $(x, y) \in \Omega$  with no-flow boundary conditions  $\mathbf{n} \cdot (D_{\text{eff}} \nabla C) = 0$  for all  $(x, y) \in \partial\Omega$ . The initial conditions are given by data in **txt** files that specify the values of  $C(x, y, 0) = C_0(x, y)$  and  $v(x, y, 0) = v_0(x, y)$  on a  $50 \times 50$  mesh of  $\bar{\Omega}$ . The physiological parameters of the problem are  $D_{\text{eff}} = 1$ ,  $\alpha = \beta = 0.1$ ,  $\gamma = 0.2$ , and  $\epsilon = 0.07$ .

The first PDE for the excitation variable is a time-dependent reaction-diffusion equation representing the excitable dynamic, and the second PDE for the recovery variable controls the local recovery of the excitation. The second PDE does not have any diffusivity term, and the  $C$ -dependence is the only reason for spatial dependence of the recovery variable. The zero-flux boundary condition is applied for the first PDE. We choose a final time of  $t_{\text{fin}} = 350$  seconds.

## 2 Use of COMSOL 4 to Solve Coupled System of PDE

In this section, we illustrate the instruction of solving a coupled system of PDE in COMSOL 4. Two approaches are used to solve the coupled system and we provide instructions for both of these approaches in Sections 2.1 and 2.2. This is followed in Section 2.3, where the methods of reading data in **txt** files and using them as initial conditions are introduced. As the last step, the post-processing of solutions including changing the appearance of plots is presented in Section 2.4.

### 2.1 Approach One: Each PDE in its own Physics Model

In this section, the step by step instructions to solve the coupled diffusion problem using the first approach are given. The idea of the first approach is to model each PDE equation in the system separately and then couple them together.

To start the model, look for **Model Wizard** in the middle panel, choose **2D, Coefficient Form PDE, Time Dependent**, and then click the **Finish** flag. To solve any system of equations it is required to create **Geometry** of domain. In our case, we create a square domain of the size  $150 \times 150$ . In the next step, the appropriate **Physics** should be added to the **Model** tab. Each equation in the coupled system of PDEs is represented by one **Physics** added to the model. We now already have one physics as **Coefficient Form PDE**, we can use it for the first equation in (1). Here the unknown is  $C(x, y, t)$ , representing the excitation variable. The right hand side  $C(C-\alpha)(1-C)-\beta v$  is specified as source term. Notice that the other variable  $v$  is coupled here. Next we need to specify the boundary condition, in this case the default zero flux boundary condition is used. Initial Condition  $C_0(x, y)$  is not constant and its value has been provided as data file. Reading initial condition to **Initial Values** of the PDE is not straightforward and is explained in the next section.

Now, to add our second equation, right click **Model 1** and select **Add Physics**, choose also **Coefficient Form PDE, Time Dependent**, and then click the **Finish** flag. We can select this second **Coefficient Form PDE (c2)** tab, and change the name of dependent variables to  $v$ , representing the recovery variable. It is straightforward to specify the source term  $\epsilon(C-\gamma v)$  and the initial condition just like for the first equation.

Finally, the mesh has been set up as **Physics-Controlled mesh**, and computation is performed to  $t_{\text{fin}} = 350$ .

## 2.2 Approach Two: Coupled PDEs in one Physics Model

In the second approach, the idea is to use only one **Physics** and represent the coupled system of PDEs by using the matrix forms of coefficients. In this example, **Physics, Coefficient Form PDE** is added to the model and two dependent variables  $C$  and  $v$  are defined, it is crucial that the coefficients are set such that they represent ((1)) correctly. In this case, **Diffusion Coefficient** is a diagonal  $2 \times 2$  matrix with 1 and 0 on diagonal. The source term is a  $2 \times 1$  vector with  $f_1 = C(C-\alpha)(1-C) - \beta v$  and  $f_2 = \epsilon(C-\gamma v)$  as sources for first and second PDE, respectively. For this example all other coefficient matrices are sparse. Again the

default zero flux boundary condition is used and the initial condition is in the form of  $2 \times 1$  vector with  $C_0(x, y)$  and  $v_0(x, y)$  as the elements.

## 2.3 Setting up Initial Conditions from Data Files

The initial condition profiles for the excitation variable,  $C_0(x, y)$ , and recovery variable,  $v_0(x, y)$ , are provided in two separate **txt** files. A single line of each of these files has three columns, containing the coordinate  $x$ , the coordinate  $y$ , and the initial value at  $(x, y)$ . In order to use these initial conditions in the system of PDEs, we create a function for each of the variables. To create these functions, right click on **Global Definitions** and select **Interpolation Function**. In the **Interpolation Function** window for **Data source** select **File** and import the appropriate **txt** file. The data format should be spreadsheet if the file is of the form described above, otherwise a different file format will be necessary. The number of arguments for this model is 2, since we have the  $x$  and  $y$  coordinates for each value. The position in file is 1 since there is only one species value in this file. Finally, **Use space coordinates as arguments** field is selected with **Mesh as Frame**.

In this way, two interpolation functions are created, one for each of the variables. In order to use these initial values in the system of PDEs, enter the name of the appropriate interpolation function as the initial value for the dependent variable in **Initial Values 1**. The initial condition profiles of excitation variable  $C$  and recovery variable  $v$  are illustrated in Figures 2 (a) and 3 (a), respectively. Figure 1 is a plot of our mesh, notice that the initial condition does not have to agree to the mesh grid, since we are using interpolation.

## 2.4 Post-Processing

The two approaches above generated identical results, allowing us to believe the same model is being built. In this section, we show how to view and output results. After the computation is done, right click on **Results** and add **2D Plot Group**. In **2D Plot Group**, we select solution set, and time that we would like to plot the result. Right click on **2D Plot Group** node and add **Surface**, in data set we choose the same solution set and use  $C$  or  $v$

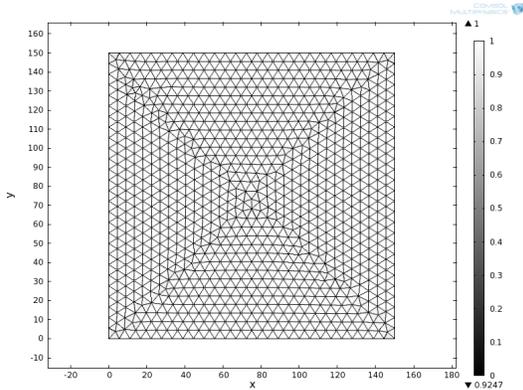


Figure 1: Mesh of the square domain.

as the expression to plot. We can manually set the range by clicking **manual color range**. To create a 3d view of the surface, we right click on **Surface** node and add **Height Expression**. In order to set the scale to real value, we use 0.01 scale factor with zero offset. We can repeat this procedure for each time step.

### 3 Results

We have demonstrated two different approaches in Section 2 to solve the coupled system of PDEs in (1), and discussed how to setup initial conditions using **txt** files. For smaller systems of coupled PDE (with two or three dependent variables), the first approach is convenient to apply. However, for larger systems which have more dependent variables of the same type, it is suggested to use the matrix form of coefficients to represent the whole system in one **Physics, Coefficient Form PDE**.

Figures 2 and 3 depict the three-dimensional view of the excitation variable and recovery variable, respectively, at time steps 0, 50, 100, ..., 350 seconds. In the initial frame, excitation is induced (interior peak), and simultaneously the corner of the domain is excited analogous to a burn line to thwart a forest fire in a given direction. Consequently, propagation of the excitation proceeds into the resting part of the domain. A recovery variable controls the local recovery of the excitation. The recovery is slow, but rapid enough for the corners of the domain adjacent to the burn line to begin to curl into the newly recovered region. Figure 4 depicts the two-dimensional view of the excitation variable, where one can easily see the curl

pattern. This physiological process is characterized by the visual appearance of a double ‘spiral wave’, a reaction-diffusion wave of the excitation variable that moves through the domain in a spiraling, recurrent way. The close resemblance of results at times  $t = 0$  and  $t = 300$  seconds as well as  $t = 50$  and  $t = 350$  seconds, in particular visible in Figure 4, bear out that the process has an approximate period of 300 seconds. This explains our choice of the final time of  $t_{\text{fin}} = 350$  seconds.

### Acknowledgments

The hardware used in the computational studies is part of the UMBC High Performance Computing Facility (HPCF). The facility is supported by the U.S. National Science Foundation through the MRI program (grant nos. CNS-0821258 and CNS-1228778) and the SCREMS program (grant no. DMS-0821311), with additional substantial support from the University of Maryland, Baltimore County (UMBC). See [www.umbc.edu/hpcf](http://www.umbc.edu/hpcf) for more information on HPCF and the projects using its resources. Co-author Huang also acknowledges financial support as HPCF RA.

### References

- [1] R. FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Bio-phys. J.*, vol. 1, pp. 445–466, 1961.
- [2] Gregory A. Handy and Bradford E. Peercy. Extending the  $IP_3$  receptor model to include competition with partial agonists. *J. Theor. Biol.*, vol. 310, pp. 97–104, 2012.
- [3] David W. Trott and Matthias K. Gobbert. Conducting finite element convergence studies using COMSOL 4.0. In Yeswanth Rao, editor, *Proceedings of the COMSOL Conference 2010, Boston, MA*, 2010.
- [4] David W. Trott and Matthias K. Gobbert. Finite element convergence studies using COMSOL 4.0a and LiveLink for MATLAB. Technical Report HPCF-2010-8, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2010.

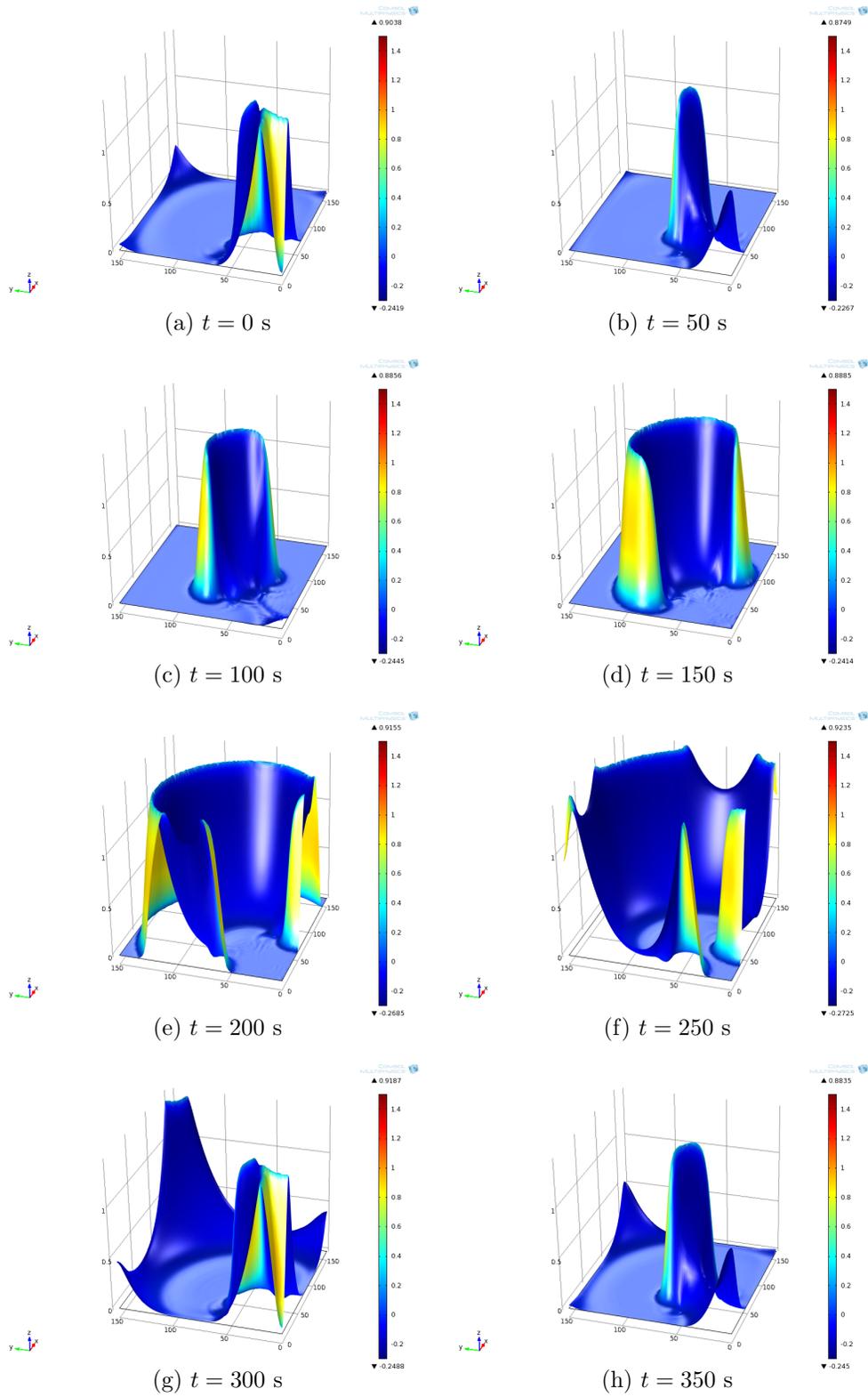


Figure 2: Three-dimensional view of the excitation variable  $C$  at different times.

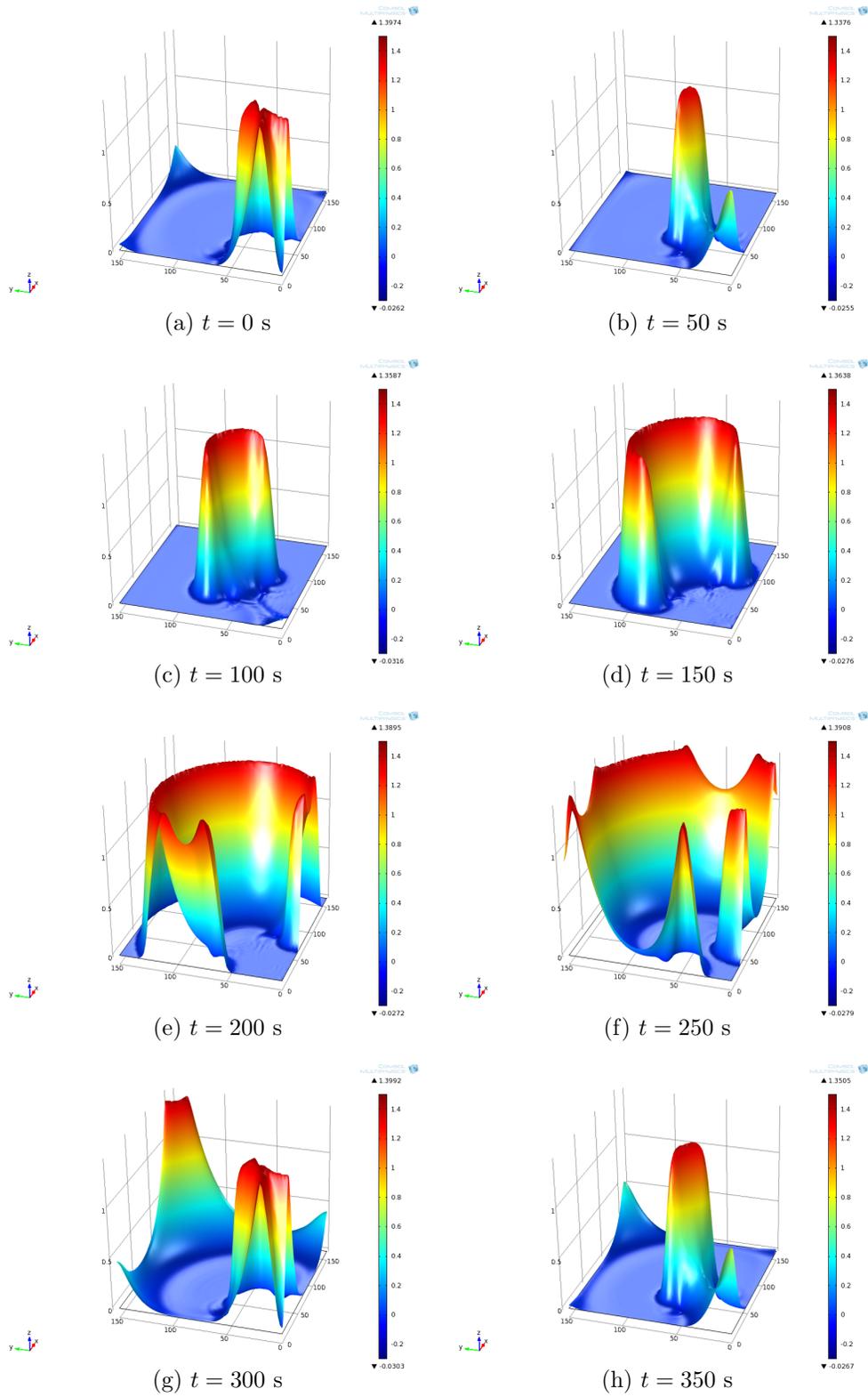


Figure 3: Three-dimensional view of the recovery variable  $v$  at different times.

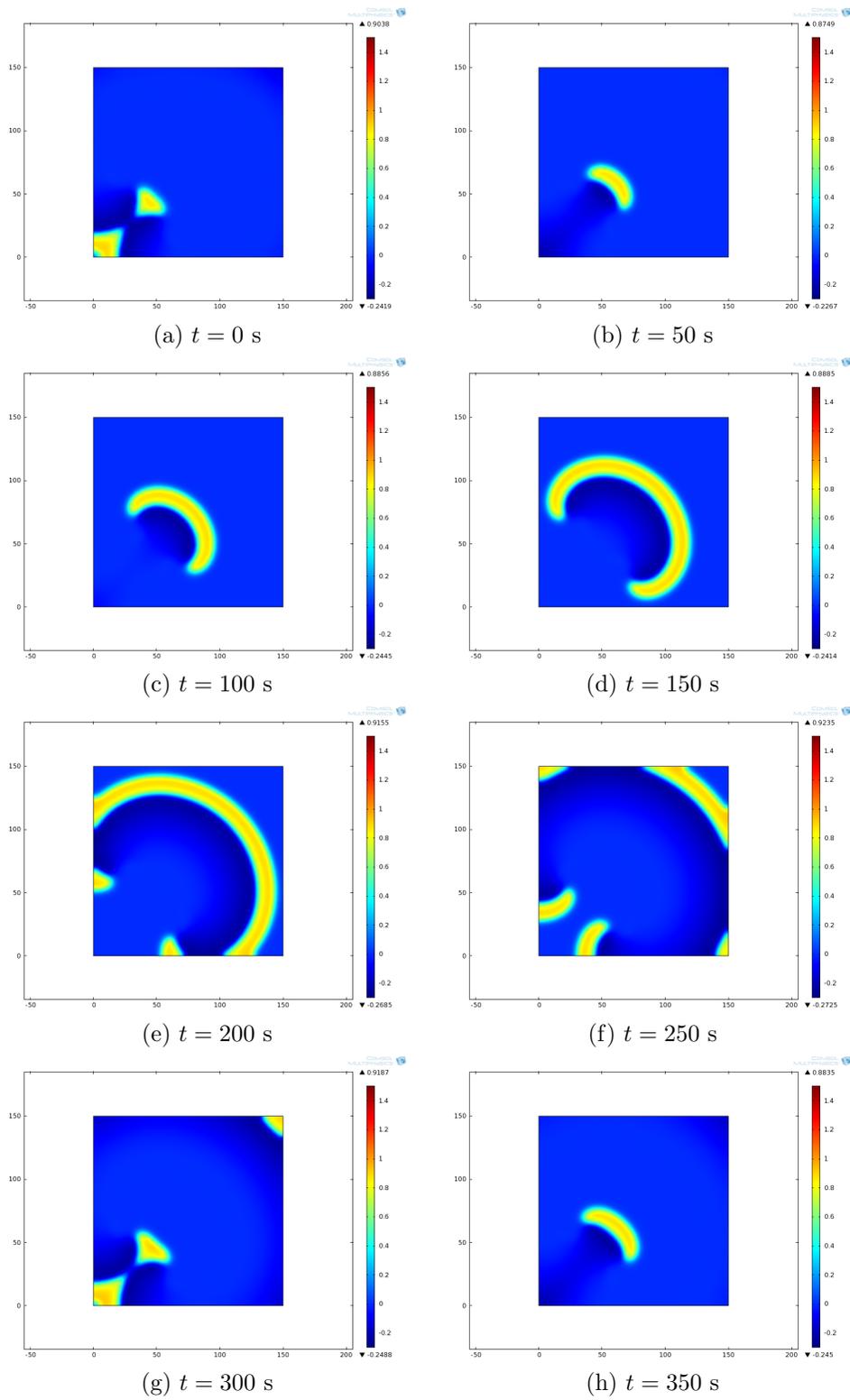


Figure 4: Two-dimensional view of the excitation variable  $C$  at different times.