COMSOL Multiphysics for Efficient Solution of a Transient Reaction-Diffusion System with Fast Reaction

Matthias K. Gobbert, Aaron Churchill, Guan Wang, and Thomas I. Seidman

Department of Mathematics and Statistics, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, gobbert@math.umbc.edu

Abstract: A reaction between chemical species is modeled by a particular reaction pathway, in which one reaction is very fast relative to the other one. The diffusion controlled reactions of these species together with a reaction intermediate are described by a system of three transient reaction-diffusion equations over a two-dimensional spatial domain. In the asymptotic limit of the reaction parameter tending to infinity, an equivalent two component model can be used in numerical simulations that is significantly more computationally efficient. But this model has features such as an unusual coupling of the two components in a boundary condition that bring out important advantages of COMSOL Multiphysics. We discuss the use of COMSOL and present representative simulation results starting with an initial condition with interesting structure throughout the domain.

Key words: Reaction-diffusion equation, Internal layer, Transient layer, Initial transient.

1 Introduction

Our objective is to study the evolution of the chemical reactions involving the two reactive species A and B, the reaction intermediate C, and a product (not tracked) modeled by the reaction pathway

$$\mathrm{A} + \mathrm{B} \xrightarrow{\lambda} \mathrm{C}, \quad \mathrm{A} + \mathrm{C} \xrightarrow{\mu} (*)$$

subject to diffusion in the two-dimensional domain $\Omega := (0,1) \times (0,1) \subset \mathbb{R}^2$. In the reaction model, λ and μ are reaction coefficients scaled such that $\lambda \gg \mu = 1$, which makes the first reaction a fast reaction relative to the second one. We denote the concentrations of the three chemically reactive species A, B, C by $u(x,y,t)$, $v(x,y,t)$, $w(x,y,t)$, respectively. Standard chemical kinetics then give a system of three transient reaction-diffusion equations for these concentrations as

$$\begin{aligned} u_t &= u_{xx} + u_{yy} - \lambda uv - uw \\ v_t &= v_{xx} + v_{yy} - \lambda uv \\ w_t &= w_{xx} + w_{yy} + \lambda uv - uw \end{aligned} \tag{1.1}$$

for $(x,y) \in \Omega$ and $0 < t \leq t_{\mathrm{fin}}$ with boundary conditions

$$\left.\begin{aligned} u &= \alpha \\ v_x &= 0 \\ w_x &= 0 \end{aligned}\right\} \text{ at } x = 0, \quad \left.\begin{aligned} u_x &= 0 \\ v &= \beta \\ w_x &= 0 \end{aligned}\right\} \text{ at } x = 1 \tag{1.2}$$

on the left and the right side of Ω, respectively, and

$$\left.\begin{aligned} u_y &= 0 \\ v_y &= 0 \\ w_y &= 0 \end{aligned}\right\} \text{ at } y = 0 \text{ and } y = 1 \tag{1.3}$$

on the top and the bottom of Ω, and the initial conditions

$$\left.\begin{aligned} u &= u_{\mathrm{ini}}(x,y) \\ v &= v_{\mathrm{ini}}(x,y) \\ w &= w_{\mathrm{ini}}(x,y) \end{aligned}\right\} \text{ at } t = 0. \tag{1.4}$$

This setup represents the case of unlimited supply of A to the left and unlimited supply of B to the right of the domain Ω.

Due to the increasingly steep gradients in the model, the numerical simulation of (1.1)–(1.4) with large values of the fast reaction parameter λ are challenging and costly. This is demonstrated in [1] for values $\lambda = 10^3$, 10^6, and particularly 10^9. However, studies for the model in one spatial dimension in [3] demonstrate that the problem is essentially in the asymptotic limit for $\lambda \geq 10^6$ and we can therefore consider the appropriate limit problem of (1.1)–(1.4) as $\lambda \to \infty$. Standard techniques

of singular perturbation analysis would give only the reduced problem $uv \equiv 0$. But [5] recently presented another approach to the problem (1.1)–(1.4) in asymptotic limit by deriving a two component model that is equivalent to the three species model. The derivation from [5] is sketched in Section 2, extended to the case of two spatial dimensions. This two component model does not have any steep gradients any more and is thus significantly cheaper computationally, as was confirmed in [1] and in [7] for the analogous problem in one spatial dimension.

However, the two component model has features such as a boundary condition that couples the (derivatives of the) solution components. This and other properties of the desired simulations make this an excellent example to bring out the benefits of several features of COMSOL Multiphysics, which are discussed in Section 3. Finally, Section 4 presents representative simulation results for the problem using the two component model as computational tool. We refer to [1] for additional plots from these simulations and to [1, 2, 7] for detailed studies that confirm the accuracy and compare the efficiency of studies based on the two component model to ones for the three species model with large values of λ.

2 Two Component Model

Computationally, the greatest difficulty in handling the system (1.1) is the occurrence of the reaction term λuv in each of the equations as $\lambda \gg 1$. Since the difference A − B and the sum B + C are each conserved in the first reaction, this difficult reaction term cancels when one combines pairs of equations in (1.1) to get

$$\begin{aligned} (u-v)_t &= (u-v)_{xx} + (u-v)_{yy} - uw \\ (v+w)_t &= (v+w)_{xx} + (v+w)_{yy} - uw \end{aligned} \tag{2.1}$$

such that the term λuv no longer appears. This motivates the transformation from u, v, w to the two components

$$\begin{aligned} u_1 &:= u - v, \\ u_2 &:= v + w. \end{aligned} \tag{2.2}$$

It is not immediately intuitive that it is possible to recover the original three variables u, v, w from the two variables u_1 and u_2. The key is to notice that in the asymptotic limit $\lambda \to \infty$, we always have the complementarity condition $uv \equiv 0$ (meaning that uv is negligible for large λ, even though, when this is multiplied by $\lambda \gg 1$, λuv can become large). Together with the facts $u \geq 0$ and $v \geq 0$ for the species concentrations u and v, we have that when $u_1 = u - v > 0$ we must have $u \neq 0$ so $v = 0$ and $u_1 = u$, while $u_1 = u - v < 0$ similarly gives $u = 0$ whence $u_1 = -v$. It then becomes possible to take

$$\begin{aligned} u &= u_1^+ := \max\{u_1, 0\}, \\ v &= -u_1^- := -\min\{u_1, 0\}, \\ w &= u_2 + u_1^-. \end{aligned} \tag{2.3}$$

The formulas in (2.2) and (2.3) in effect constitute forward and backward transformations between the three species and the two component models and make the two models equivalent to each other [5].

The equivalent two component model is then

$$\begin{aligned} u_{1,t} &= u_{1,xx} + u_{1,yy} - u_1^+ u_2 \\ u_{2,t} &= u_{2,xx} + u_{2,yy} - u_1^+ u_2 \end{aligned} \tag{2.4}$$

for $(x, y) \in \Omega$ and $0 < t \leq t_{\text{fin}}$ with boundary conditions

$$\left.\begin{aligned} u_1 &= \alpha \\ u_{2,x} &= 0 \end{aligned}\right\} \text{at } x = 0, \quad \left.\begin{aligned} u_1 &= -\beta \\ u_{2,x} &= -u_{1,x} \end{aligned}\right\} \text{at } x = 1 \tag{2.5}$$

on the left and the right side of Ω, respectively, and

$$\left.\begin{aligned} u_{1,y} &= 0 \\ u_{2,y} &= 0 \end{aligned}\right\} \text{ at } y = 0 \text{ and } y = 1, \tag{2.6}$$

on the top and the bottom of Ω, and initial conditions

$$\left.\begin{aligned} u_1 &= u_{1,\text{ini}}(x, y) \\ u_2 &= u_{2,\text{ini}}(x, y) \end{aligned}\right\} \text{ at } t = 0. \tag{2.7}$$

Most transformations of the boundary conditions and initial conditions follow directly from the definitions in (2.2); to see the second boundary condition at $x = 1$, notice that $u_{2,x} = v_x + w_x = v_x$, since $w_x = 0$ at $x = 1$. Using $u_1 = u_1^+ + u_1^-$ gives $u_{1,x} = u_{1,x}^+ + u_{1,x}^- = u_x - v_x$. Since $u_x = 0$ at $x = 1$, we thus have $u_{1,x} = -v_x$ and the boundary condition $u_{2,x} = v_x = -u_{1,x}$ at $x = 1$.

The complete two component system (2.4)–(2.7) is self-contained and computationally efficient, since it does not have any steep gradients [1]. The price is a rather unusual coupling in the second boundary condition at $x = 1$.

3 Use of COMSOL

The computations use COMSOL Multiphysics 3.5a on the cluster hpc in the UMBC High Performance Computing Facility (`www.umbc.edu/hpcf`). COMSOL is run on one node with two dual-core AMD Opteron processors (2.6 GHz, 1 MB cache per core) and 13 GB of memory. The cluster runs the Linux RedHat EL5 operating system. Several features of the problem under consideration serve to highlight the features of COMSOL:

- One of the crucial features of the reaction--diffusion equations under consideration are their non-linear reaction terms. Since the physics of this problem relies on a subtle balance between the diffusion effects and these reactions, we wish to represent them as accurately as possible. Thus, we use the General Form of the PDEs in COMSOL in order to enable COMSOL to compute symbolic derivatives of all terms in the PDEs by automatic differentiation for the highest accuracy in the evaluation of the Jacobian in the non-linear solver inside the implicit ODE method. By contrast, the Coefficient Form in COMSOL approximates the derivatives of all terms numerically, which is only suitable for mildly non-linear PDEs.

- An interesting feature of the two component problem is the boundary condition $u_{2,x} = -u_{1,x}$ at $x = 1$ that couples the (derivatives of the) solution components. Many PDE solvers have difficulty handling a boundary condition of this type or do not permit specifying it at all, even for the analogous problem in only one spatial dimension. COMSOL has no trouble with this, as all terms specified are allowed to include also the dependent variables and their derivatives.

- We couple COMSOL with Matlab using the Linux command `comsol matlab`. With this, a script is written so that each computation would be using the same solver parameters inside COMSOL. This is crucial to ensure reproducibility of all results and useful to facilitate parameter studies, by that with respect to problem parameters or with respect to numerical parameters. Scripting is also very useful in cases such as the two component model, where significant post-processing such as back-transforming to the original variables by (2.3) is necessary. But we use the flexibility of scripting even more for the problems under consideration here. COMSOL easily allows the user to specify functional expressions for boundary and initial conditions. This is suitable for the boundary conditions here, and changing one of them is possible in the script. But the initial conditions desired here will have a fairly complicated structure inside the domain, and we moreover wish to make changing the initial condition convenient. Using functional expressions in the COMSOL script becomes too cumbersome and potentially limiting. Therefore, we merely specify the names of functions for the initial conditions in the script, which enables the user to write these functions outside of COMSOL using the full capabilities of high-level programming languages such as if-statements, for-loops, and vector operations on the vectors of (x, y) values.

- We use linear Lagrange finite elements on a uniform quadrilateral mesh with $N \times N$ elements. The studies below report results from a 128×128 mesh, but also studies with 64×64 and 256×256 elements were performed [1]. At the coarser mesh resolutions, some of the geometric features are not as accurate as desired. The finer mesh did not add any appreciable detail for the particular initial conditions considered here. Clearly, computations increase significantly in complexity, hence we do not wish unnecessarily large values of N in production runs. Note that the defaults in COMSOL are an unstructured mesh and quadratic Lagrange finite elements. By using a structured, uniform, quadrilateral mesh, we avoid any incidental biasing that might affect the solution as result of, e.g., from element boundaries being at different random non-horizontal or non-vertical angles or similar properties inherent to an unstructured mesh. Because we will be interested in the exact location of interfaces later, it is important to avoid any such biasing for this problem. Because the solutions have jump discontinuities at the initial time and will have discontinuities in their derivatives at the reaction interfaces [5], we use the lowest order Lagrange finite elements available in COMSOL.

- COMSOL offers several ODE solvers, all of which use implicit time stepping, which is a necessity for efficiently solving PDEs of parabolic type such as reaction-diffusion equations. Since we desire to compute to a large final time approaching steady state, sophisticated ODE solver features such as automatic time stepping and method order selection up to a high order are vital. This becomes particularly clear if you consider that we have to expect steep initial gradients from the chosen initial conditions, which requires small time steps and low ODE method orders, but that we wish to use large time steps and high method orders, when the solution is smooth in its approach to steady state. The ODE solver used is BDF-DASPK. We use a relative tolerance of 10^{-3} and an absolute tolerance of 10^{-6} for the local error control in the ODE solver, which is slightly tighter than COMSOL defaults. We experimented with coarser as well as tighter tolerances. At coarser tolerances, some features of the solution are not as clear. Tighter tolerances confirm the results obtained for the tolerances used, thus these are the most effective tolerances to use. The simulations reported below using the two component model required 229 time steps and took 95 seconds. This is markedly faster than simulations for the three species model with large λ values, which can be, say, 1,743 time steps and 1,802 seconds for $\lambda = 10^9$ [1, Table 1]. The increased numerical difficulty associated with such large λ values materializes also in other ways, e.g., that the ODE solver broke down for the coarse 64×64 mesh at an intermediate time and we were forced to use a coarser ODE tolerance for this case. We choose the ODE solver BDF-DASPK because BDF-IDA, the default solver, has trouble converging at the initial conditions for the case of large λ values. The ODE solver Generalized Alpha was also experimented with, but this too had some difficulties.

- We use the linear solver PARDISO, since it is supposed to profit most from the multithreading available on the dual-core processors used; see [4] in the same proceedings as this paper.

4 Results

In this section, we present representative numerical results in the form of surface and contour plots obtained by COMSOL Multiphysics. These simulations use the two component model (2.4)–(2.7) to efficiently compute u_1 and u_2, and then we back-transform to the original variables u, v, w of (1.1)–(1.4) using (2.3).

We are interested in initial condition functions u_{ini}, v_{ini}, w_{ini} such as shown in Figure 1, where each connected piece of each initial concentration has some constant value. For u in Figure 1 (a), which has an unlimited supply with concentration $\alpha = 1.6$ at $x = 0$, the left hand portion of the domain, all portions of the domain connected to it, as well as the small disjoint disk in the upper right of the domain have values of $u_{\text{ini}} = \alpha$, and $u_{\text{ini}} = 0$ everywhere else. For v in Figure 1 (b), which has an unlimited supply with concentration $\beta = 0.8$ at $x = 1$, the values are $v_{\text{ini}} = 0$ wherever $u_{\text{ini}} > 0$ and $w_{\text{ini}} = \beta$ wherever $u_{\text{ini}} = 0$. We note that by this construction the product $uv \equiv 0$ at $t = 0$. The concentration of w_{ini} is 0 throughout the domain at $t = 0$ and not shown in the figure. The fast reaction in the reaction model is restricted to areas of Ω where u and v co-exist. This is only the case along the interface through the domain where u and v come in contact due to diffusion. The locations of these interface lines are shown in Figure 1 (c). The interface location is determined numerically as the 0 level of a contour plot of the quantity $u - v$ (with only this one contour level shown in the plot). To allow us a concrete reference to various parts of the domain, we use the following terminology: The portion on the left side of the domain is called the 'body,' and the piece protruding from the body is called the 'head,' which is connected to the body by the 'neck.' The inspiration for these terms is most evident in the interface plot in Figure 1 (c).

Figures 1, 2, 3, and 4 each show the concentrations u and v and the interface where u and v come in contact due to diffusion, at times 0, 10^{-3}, 10^{-2}, and 20, respectively. After starting with sharp interfaces between areas with positive u or v and 0 in Figure 1, there is a rapid initial transient in time during which the reaction-diffusion equations smooth out the concentrations. This effect is visible in Figures 2 and 3 for u and v. The most intriguing feature of the model under study is the behavior resulting for the location of the reaction

interface, where $u > 0$ and $v > 0$ meet by diffusion and react rapidly. Recall that at the initial time, these two species do not coexist and mathematically $uv \equiv 0$. It is only by diffusion that positive values of both get in contact with each other and this defines the reaction interface. At the initial time $t = 0$ in Figure 1 (c), we can still clearly see the 'head' connected by the 'neck' to the 'body' on the left portion of the domain and an disjoint 'disk' in the upper right of the domain. By $t = 10^{-3}$ in Figure 2 (c), we see the head separating from the body in the left portion of the domain, before it re-attaches to the body by $t = 10^{-2}$ in Figure 3 (c). Also by the time of Figure 3, the disjointed disk in the upper right of the domain has vanished with its supply of the u species completely consumed. By $t = 20$ in Figure 4, the solution of the problem has reached its steady state; this is confirmed by comparing the solutions for u, v, and w as functions of $0 < x < 1$ for each y to the solution of the stationary problem (with $u_t = v_t = w_t = 0$) associated with (1.1)–(1.2) in one spatial dimension [6]. Numerically, we can moreover compare the location of the interface in Figure 4 (c) to its known steady state location at $x^* \approx 0.6$; see [7, Table 1].

The chosen initial conditions give rise to an interesting behavior in that the head separates and then re-attaches to the body in the left portion of the domain, while the disjoint disk disappears. Additional plots for this simulation based on the two component model are shown in [1], along with comparisons to studies with the three species model for several values of λ with respect to accuracy and efficiency.

Acknowledgments

The work reported in this paper is based on consulting projects in the Center for Interdisciplinary Research and Consulting (`www.umbc.edu/circ`). 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 no. CNS–0821258) 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` for more information on HPCF and the projects using its resources.

References

[1] Aaron Churchill, Matthias K. Gobbert, and Thomas I. Seidman. Efficient computation for a reaction-diffusion system with a fast reaction in two spatial dimensions using COMSOL Multiphysics. Technical Report HPCF–2009–7, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2009.

[2] Aaron Churchill, Guan Wang, Matthias K. Gobbert, and Thomas I. Seidman. Efficient simulation of a reaction-diffusion system with a fast reaction in the asymptotic limit. In preparation.

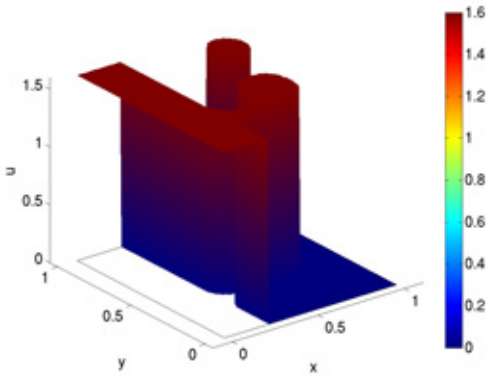
[3] Michael Muscedere and Matthias K. Gobbert. Parameter study of a reaction-diffusion system near the reactant coefficient asymptotic limit. *Dynamics of Continuous, Discrete and Impulsive Systems Series A Supplement*, pages 29–36, 2009.

[4] Noemi Petra and Matthias K. Gobbert. Parallel performance studies for COMSOL Multiphysics using scripting and batch processing. In Yeswanth Rao, editor, *Proceedings of the COMSOL Conference 2009, Boston, MA*, 2009.

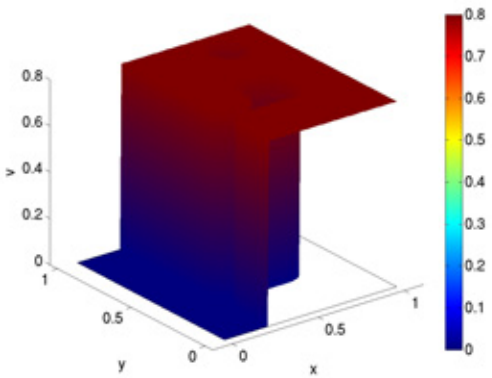
[5] Thomas I. Seidman. Interface conditions for a singular reaction-diffusion system. *Discrete and Cont. Dynamical Systems*, to appear.

[6] Ana Maria Soane, Matthias K. Gobbert, and Thomas I. Seidman. Numerical exploration of a system of reaction-diffusion equations with internal and transient layers. *Nonlinear Anal.: Real World Appl.*, 6(5):914–934, 2005.

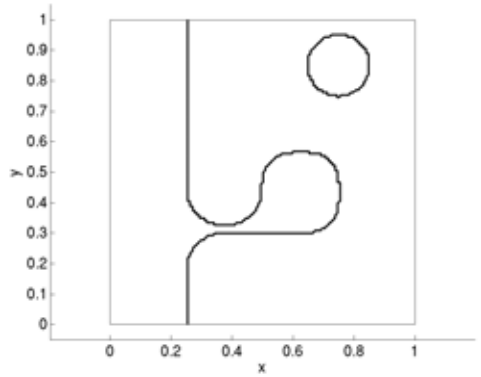
[7] Guan Wang, Aaron Churchill, Matthias K. Gobbert, and Thomas I. Seidman. Efficient computation for a reaction-diffusion system with a fast reaction with continuous and discontinuous initial data using COMSOL Multiphysics. Technical Report HPCF–2009–3, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2009.



(a) $u(x, y)$ vs. (x, y)

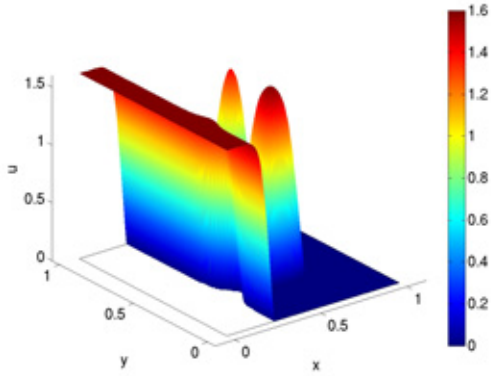


(b) $v(x, y)$ vs. (x, y)

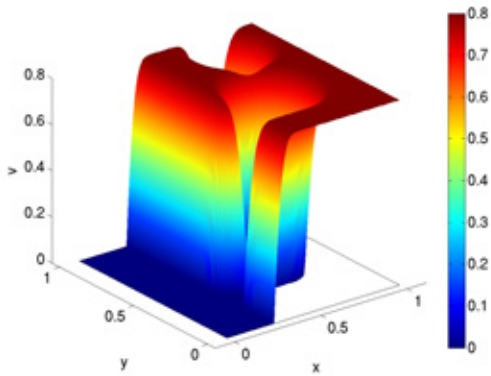


(c) interface vs. (x, y)

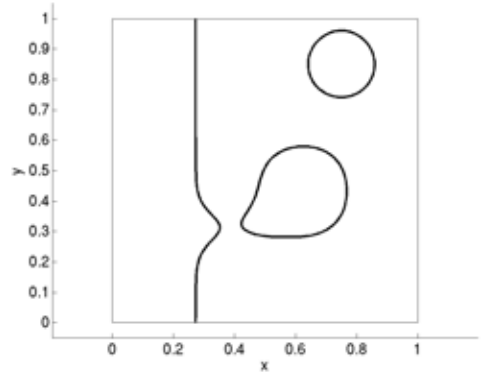
Figure 1: u, v, and the interface at $t = 0$.



(a) $u(x, y)$ vs. (x, y)

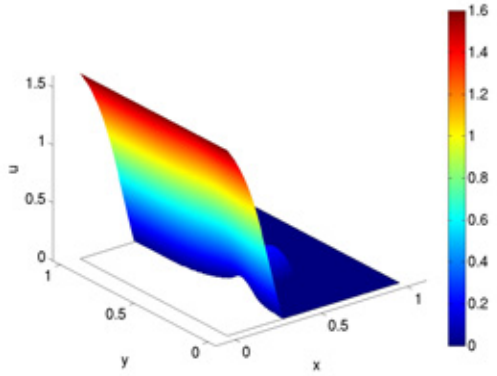


(b) $v(x, y)$ vs. (x, y)

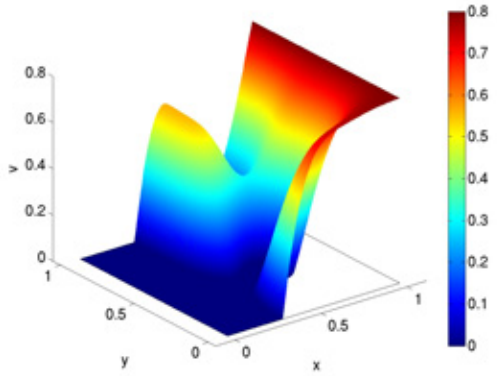


(c) interface vs. (x, y)

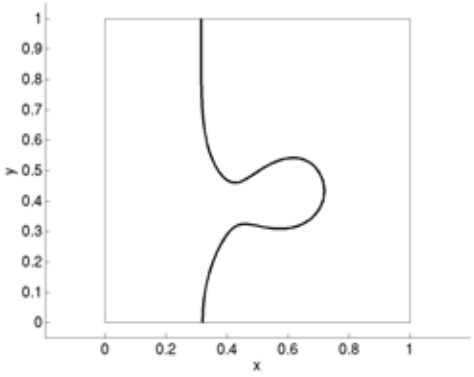
Figure 2: u, v, and the interface at $t = 10^{-3}$.



(a) $u(x, y)$ vs. (x, y)

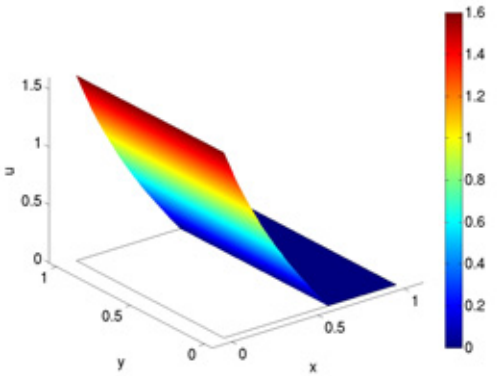


(b) $v(x, y)$ vs. (x, y)

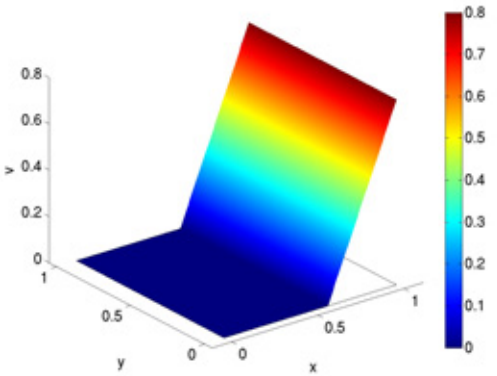


(c) interface vs. (x, y)

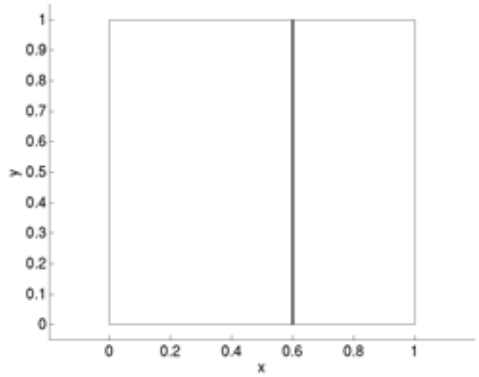
Figure 3: u, v, and the interface at $t = 10^{-2}$.



(a) $u(x, y)$ vs. (x, y)



(b) $v(x, y)$ vs. (x, y)



(c) interface vs. (x, y)

Figure 4: u, v, and the interface at $t = 20$.