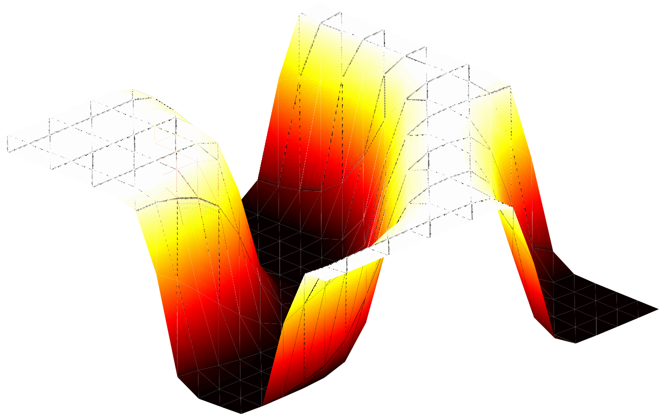
Reaction–diffusion

Computer Session E4



Background

Today, we will solve the *bistable* equation,

$$\begin{aligned} \dot{u} - \nabla \cdot (a \nabla u) &= u(1-u^2) \quad \text{in } \Omega \times (0,T], \\ -\partial_n u &= 0 \quad \text{on } \Gamma \times (0,T], \\ u(\cdot,0) &= u_0 \quad \text{in } \Omega, \end{aligned} \tag{1}$$

where we note that the right-hand side depends on the solution u itself. This is a *nonlinear* equation (including terms with both u^2 and u^3). We call this a *reaction–diffusion* equation, since it contains both the diffusive term $-\nabla \cdot (a \nabla u)$ and a term depending only on u, the reaction term $u(1-u^2)$.

The dG(0) formulation of the bistable equation is given by

$$\int_\Omega U_n v\,dx + k\int_\Omega a\nabla U_n\cdot\nabla v\,dx = \quad k\int_\Omega U_n(1-U_n^2)v\,dx + \int_\Omega U_{n-1}v\,dx \quad \forall v\in V_h, \tag{2}$$

where $k = t_n - t_{n-1}$ denotes the size of the time step, U_n denotes the value at $t = t_n$, and U_{n-1} denotes the value at $t = t_{n-1}$.

Before today's computer session, make sure that you understand and can answer the following questions.

Question 1 Derive the variational formulation (2) from the bistable equation (1).

Question 2 How does the corresponding variational formulation look for the cG(1) method? This method is also known as the Crank-Nicolson method.

Question 3 Try to find three stationary solutions of the bistable equation. Hint: For which values of u is the right-hand side zero? Which of these stationary solutions are stable, i.e., if we perturb the stationary solution slightly, will it then try to go back to the stationary state or will it continue to change? Why do you think the equation is called the *bi*stable equation?

Preparations

Create a new directory called `e4` and the two subdirectories `problem1` and `problem2`. Then download the following files to each of the subdirectories:

- `AssembleMatrix.m`,
- `AssembleVector.m`,
- `Bistable.m`,
- `makemovie.m`,
- the files in the directory `mesh`.

Also download the file `solution1.mat` to the directory `problem1` and the file `solution2.mat` to the directory `problem2`.

These files are available on the web page of this session under *Programs and templates*.

Problems

Problem 1

Preparation

Go to the directory `problem1`. Create a file called `BistableSolver.m` and write your program in this file. You also have to edit the file `Bistable.m`, where you specify the variational formulation.

Problem

Solve the bistable equation on the unit square (`square.m`) with homogeneous Neumann boundary conditions, $a = 0.01$, and $T = 20$. Use a time step of size $k = 0.1$ and let the initial condition be given by

$$u_0(x) = \cos(2\pi x_1^2)\cos(2\pi x_2^2). \tag{3}$$

To solve the problem, look at your solver for the time-dependent convection–diffusion equation from computer session E3, or try yourself without looking.

Since the problem is now nonlinear, we need to use fixed-point iteration. When we solved the (linear) convection–diffusion equation, we only needed to assemble the vector `b` and then solve the linear system `A U1 = b` in each time step. Now, we need to solve the nonlinear system

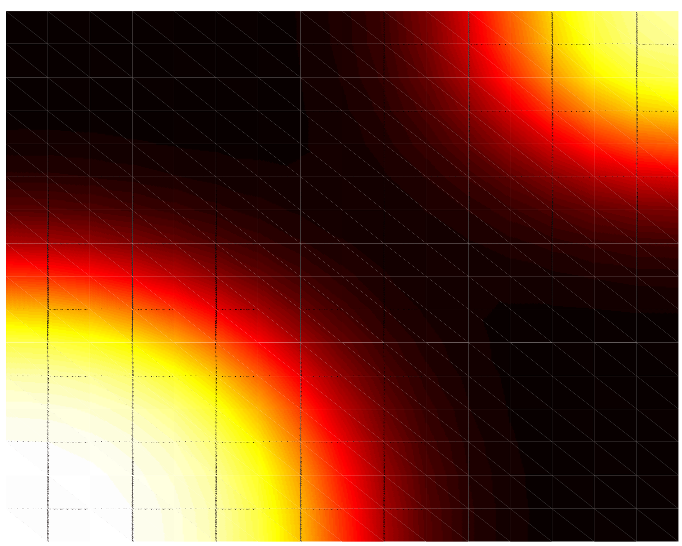
```
A U1 = b(U1)
```

in each time step using fixed-point iteration. We do this by repeatedly assembling the vector `b` for a given value of `U1` and then updating the value of `U1` by solving the linear system.

In each iteration, the latest value of `U1` is given to the assembler using the optional argument `W`. Before, we used `W` only to represent `U0` (which becomes `w(1)` in the variational formulation), but now we will use `W = [U0 U1]` for both `U0` and `U1`. In the variational formulation, we can then use `w(1)` to obtain the value of `U0` and `w(2)` to obtain the value of `U1`.

Plot the solution using the `pdesurf` command in each time step. Remember to write `pause` after each plot. You could also use the commands `view(2)` (to see the solution from above) and `colormap jet` (for some nice colors).

Check your answer: Compare your solution with a reference solution using the commands `load solution1` and then `makemovie`. This solution has been computed on a finer mesh (`square_refined.m`) with a time step of size $k = 0.1$ and $T = 20$.



Problem 2

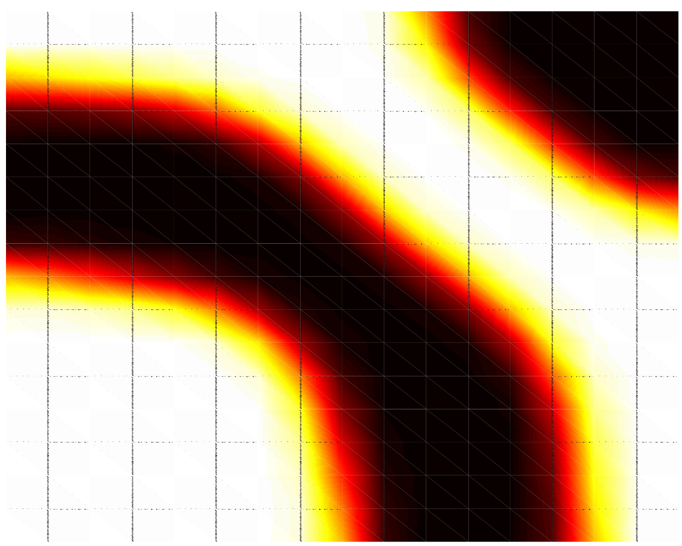
Preparation

Go to the directory `problem2` and copy the two files `BistableSolver.m` and `Bistable.m` from the directory `problem1`.

Problem

Modify your program from Problem 1 to use a diffusivity of size $a = 0.001$ and compute with the same time step until time $T = 20$. Compare to your solution in Problem 1. Can you explain the difference? What happens when we decrease the diffusion?

Check your answer: Compare your solution with a reference solution using the commands `load solution2` and then `makemovie`. This solution has been computed on a finer mesh (`square_refined.m`) with a time step of size $k = 0.1$ and $T = 20$.



Hints

Problem 1

The variational formulation that you need to specify in the file `Bistable.m` is given by

```
u*v*dx + k*0.01*du'*dv*dx
```

for the left-hand side and

```
k*w(2)*(1 - w(2)*w(2))*v*dx + w(1)*v*dx
```

for the right-hand side.

The fixed-point iteration can be implemented as follows:

```
while 1

  % Assemble vector
  b = AssembleVector(p, e, t, 'Bistable', [U0 U1], time);

  % Solve the linear system
  newU1 = A \ b;

  % Check if the solution has converged
  if norm(newU1 - U1) < 0.01
    break;
  end

  % Update U1 to new value
  U1 = newU1;

end
```

Problem 2

Since we decrease the value of a, large gradients (thin layers) need to form before the diffusion will start to work. Remember that the diffusion term is given by

$$-\nabla \cdot (a\nabla u), \tag{4}$$

and so a large gradient ∇u can compensate for a small diffusivity a.

Solutions

Make sure that you really try to solve each problem before looking at the solutions. Have you really tried to solve the problem or should you try again before looking at the solution?

The solutions are available on the web page of this session under *Solutions to problems.*

About

This Computer Session is part of the Body and Soul educational program. More information can be found at

`http://www.phi.chalmers.se/bodysoul/`

This Computer session is written by Anders Logg.