

[CAD CAE] Lab 5

Oryginalne zdjęcie



Pierwszy gif - wybuchł

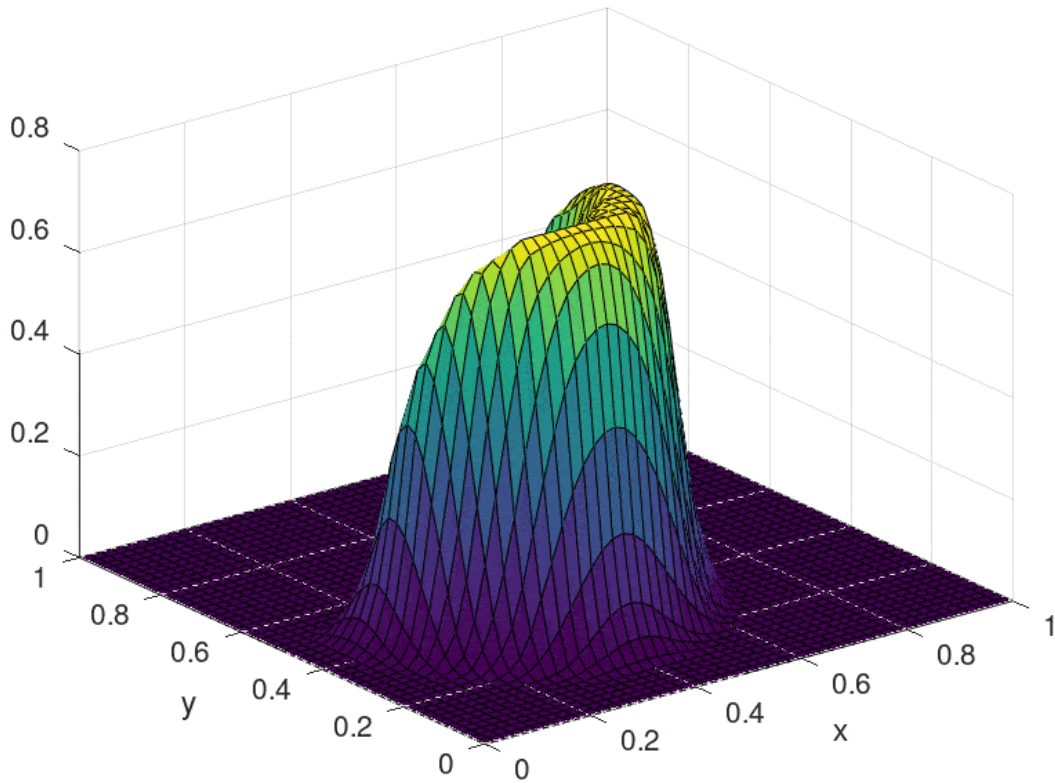
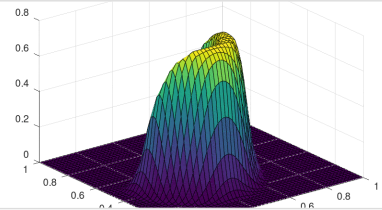
```
knot = simple_knot(20 , 2);  
dt = 0.0003;  
theta = 0;  
K = 30;
```

Link: <https://gifyu.com/image/S27ng>

gifed

Image gifed hosted in Gifyu

<https://gifyu.com/image/S27ng>



Drugi gif

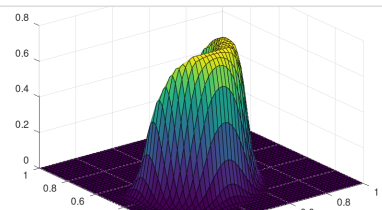
```
knot = simple_knot(20 , 2);  
dt = 0.00005;  
theta = 0;  
K = 100;
```

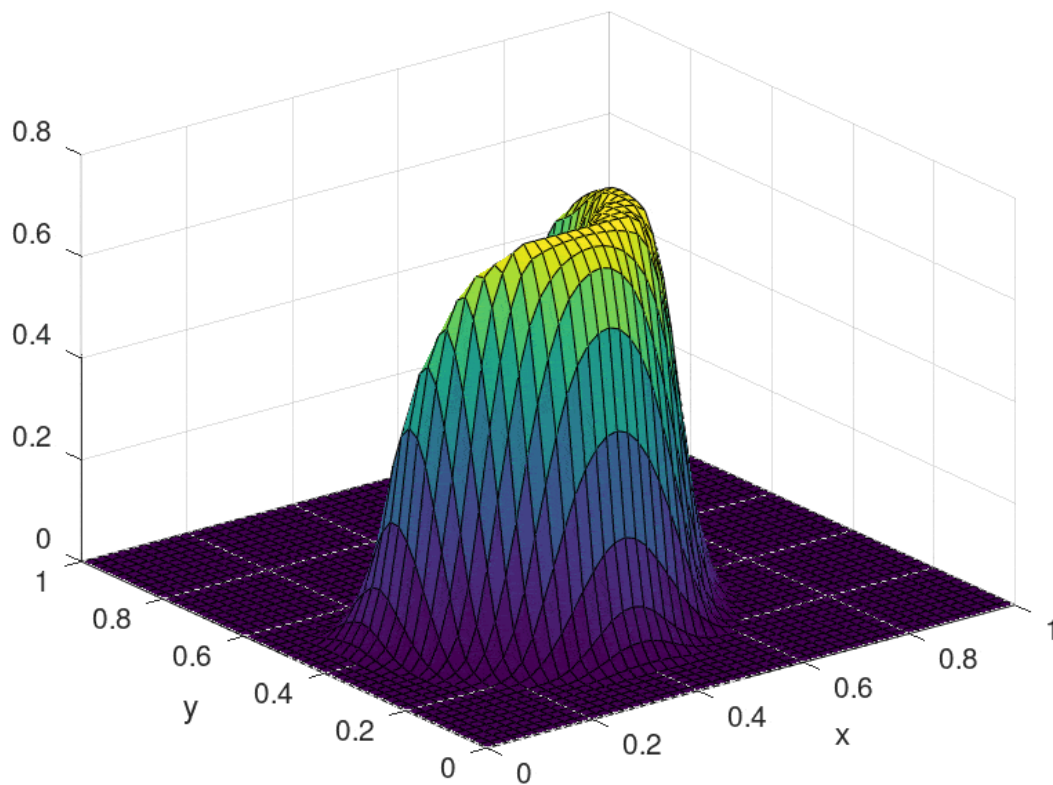
Link: <https://gifyu.com/image/S27u0>

gifed

Image gifed hosted in Gifyu

<https://gifyu.com/image/S27u0>





Zmodyfikowany kod

```
% Input data
knot = simple_knot(20 , 2)      % knot vector (5,2)
dt = 0.00005;                  % time step size
theta = 0;                     % scheme parameter (0 - explicit Euler, 1 - implicit Euler, 1/2 - Crank-Nicolson)
K = 100;                        % number of time steps

% Problem formulation
%f = @(t, x) 1;
%init_state = @(x) 0;

f = @(t, x) 0;
init_state = @(x) init_state_1(x);

% Setup
p = degree_from_knot(knot);
k = p + 1;

points = linspace(0, 1, max(knot) + 1);

bx = basis1d(p, points, knot);
by = basis1d(p, points, knot);

nx = number_of_dofs(bx);
ny = number_of_dofs(by);
n = nx * ny;

M = sparse(n, n);
F = zeros(n, 1);

idx = @(dof) linear_index(dof, bx, by);
```

```

% Assemble the matrix
for e = elements(bx, by)
    J = jacobian2d(e, bx, by);
    for q = quad_data2d(e, k, bx, by)
        basis = basis_evaluator2d(q.x, bx, by);

        for i = dofs_on_element2d(e, bx, by)
            [v, dv] = basis(i);
            for j = dofs_on_element2d(e, bx, by)
                [u, du] = basis(j);
                val = u * v + dt * theta * dot(du, dv) + dt*u*(1-u)*v;
                M(idx(i), idx(j)) = M(idx(i), idx(j)) + val * q.w * J;
            end
        end
    end
end

% Modify the matrix to account for uniform Dirichlet boundary conditions
fixed_dofs = boundary_dofs2d(bx, by);
[M, F] = dirichlet_bc_uniform(M, F, fixed_dofs, bx, by);

data = imread("img_center_small2_inv.png");

tmp = cast(data, 'double') / cast(320, 'double');
tmp = [tmp; zeros(20, 100, 3)];
tmp = [tmp; zeros(50, 120, 3)];

u = tmp

% Put the initial state into u
%u = project2d(init_state, bx, by)

% Plot the initial state
save_plot(u, 0, bx, by);

```