

---

```
% Homework 10
clc;
clear;
close all;

f = @(x, y) (x+y)./(1 + 25*(x.^2)+y.^2));

a = -1;
b = 1;

n = (5:5:30)';
n_eval = 100;

equispaced_nodes_errs = zeros(2, length(n));
chebyshev_nodes_errs = zeros(2, length(n));

for i = 1:length(n)
    n_i = n(i);
    % x_nodes = linspace(a, b, n_i);
    x_nodes = cos((2*(0:n_i-1)+1)*pi/(2*(n_i)));
    y_nodes = x_nodes;
    [X, Y] = meshgrid(x_nodes, y_nodes);
    x = reshape(X, 1, (n_i)^2);
    y = reshape(Y, 1, (n_i)^2);
    f_xy_nodes = f(x, y);

    % Create the tensor grid Lagrange interpolant
    x_eval = linspace(a, b, n_eval);
    y_eval = x_eval;
    [XX, YY] = meshgrid(x_eval, y_eval);
    xx_eval = reshape(XX, 1, n_eval^2);
    yy_eval = reshape(YY, 1, n_eval^2);
    xy_eval = [xx_eval; yy_eval];
```

## Regular tensor Lagrange interpolant

```
z = tensorproduct2D_lagrange(xy_eval, x_nodes, y_nodes, f_xy_nodes);
Z = reshape(z, n_eval, n_eval);

if n_i == 20
    figure;
    surf(XX, YY, Z);
    xlabel("x");
    ylabel("y");
    title("Plot of regular tensor Lagrange interpolant for n = 20");
end
```

## Boolean sum Lagrange interpolant

```
s_x = zeros(length(xx_eval), 1);
for k = 1:n_i
```

---

```

        s_x = s_x + f(x_nodes(k), yy_eval').*eval_lagrange(xx_eval', x_nodes,
k);
    end

    s_y = zeros(length(yy_eval), 1);
    for k = 1:n_i
        s_y = s_y + f(xx_eval', y_nodes(k)).*eval_lagrange(yy_eval', y_nodes,
k);
    end

    s = reshape(s_x + s_y, n_eval, n_eval) - Z;

    if n_i == 20
        figure;
        surf(XX, YY, s);
        xlabel("x");
        ylabel("y");
        title("Plot of Boolean sum Lagrange interpolant for n = 20");
    end

    % Display the error (in the 2-norm) between the interpolant and actual
    % function values
    f_xy_eval = f(xx_eval, yy_eval);
    % equispaced_nodes_errs(1, i) = norm(f_xy_eval - z');
    % equispaced_nodes_errs(2, i) = norm(f_xy_eval - reshape(s, 1, n_eval^2));
    chebyshev_nodes_errs(1, i) = norm(f_xy_eval - z');
    chebyshev_nodes_errs(2, i) = norm(f_xy_eval - reshape(s, 1, n_eval^2));
end

figure;
semilogy(n, chebyshev_nodes_errs(1, :), "LineWidth", 2, "DisplayName", "Tensor
product 2-norm errors");
hold on;
semilogy(n, chebyshev_nodes_errs(2, :), "LineWidth",
2, "DisplayName", "Boolean sum Lagrange 2-norm erros");
xlabel("n");
ylabel("2-norm of error between f and approximation");
% title("2-norm of errors between f and approximations for various n using
equispaced nodes");
title("2-norm of errors between f and approximations for various n using
Chebyshev nodes");
legend;

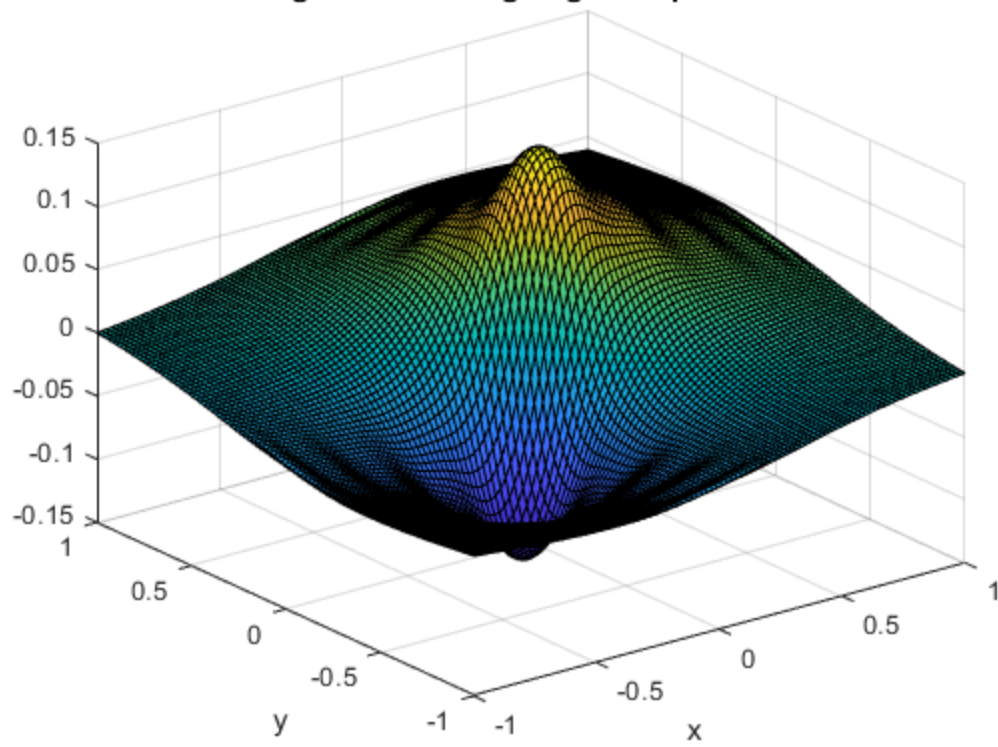
function val = eval_lagrange(x, x_nodes, j)
    val = ones(length(x), 1);
    for i = 1:length(x_nodes)
        if i ~= j
            val = val .* (x-x_nodes(i))/(x_nodes(j)-x_nodes(i));
        end
    end
end
end

```

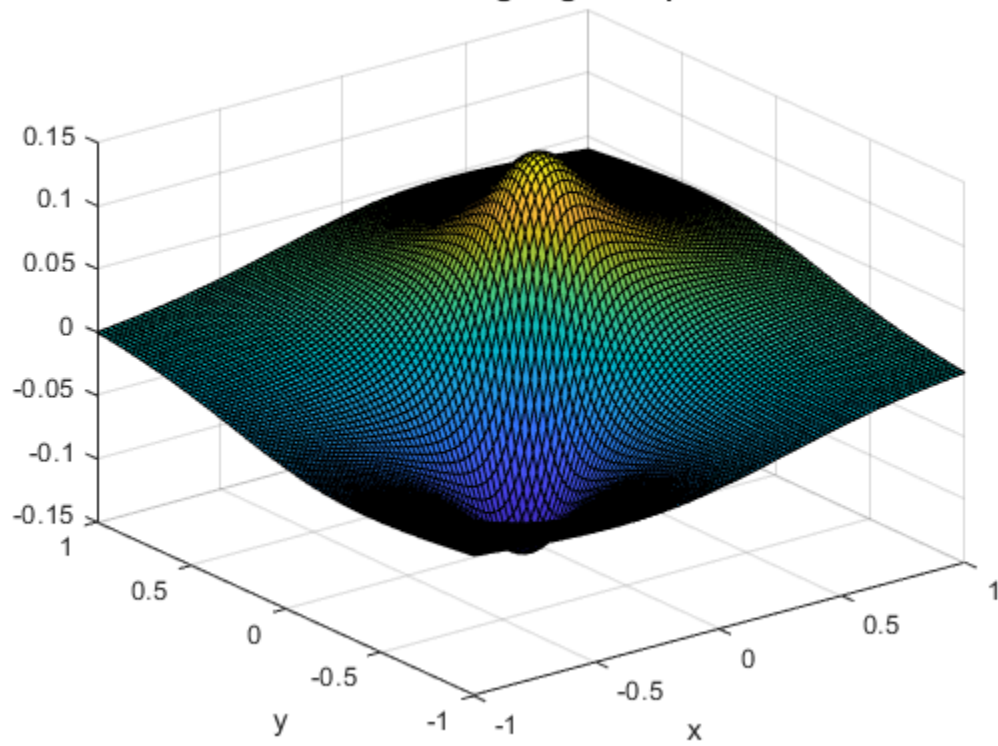
---

---

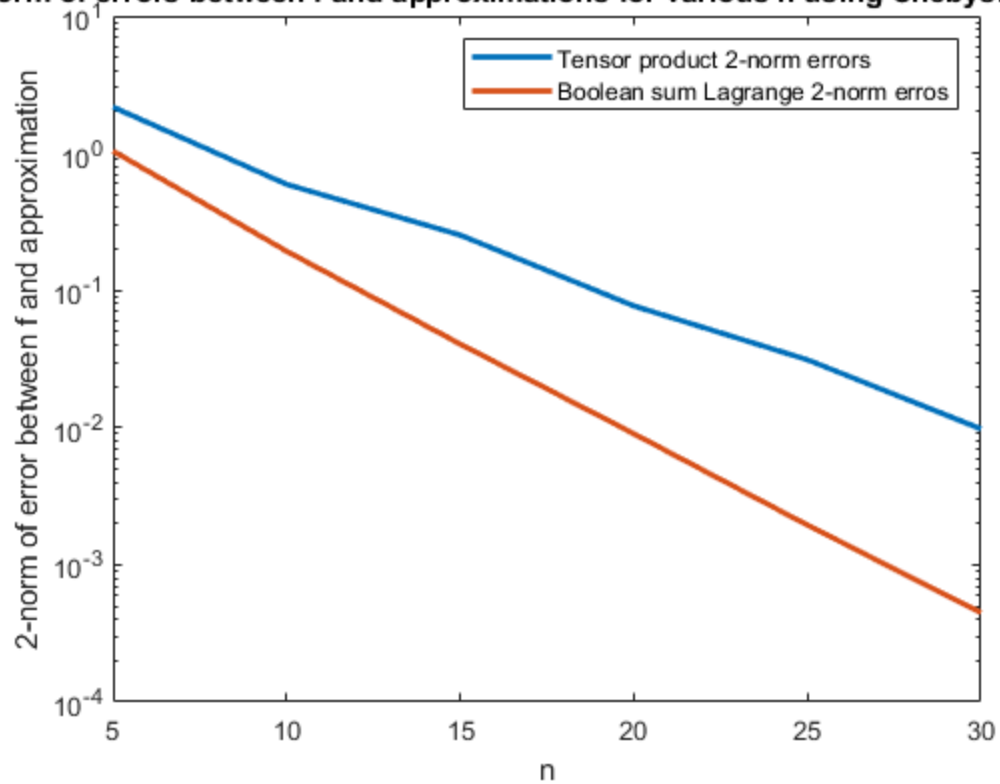
**Plot of regular tensor Lagrange interpolant for  $n = 20$**



**Plot of Boolean sum Lagrange interpolant for  $n = 20$**



2-norm of errors between f and approximations for various n using Chebyshev nodes



Published with MATLAB® R2021b