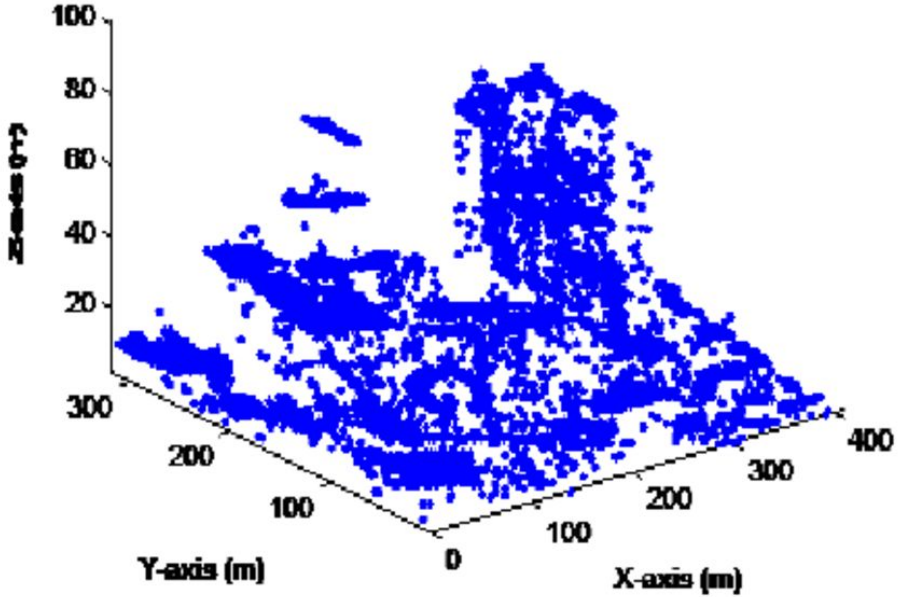
Model of terrain using RBFs

Mateusz Wszeborowski 165562

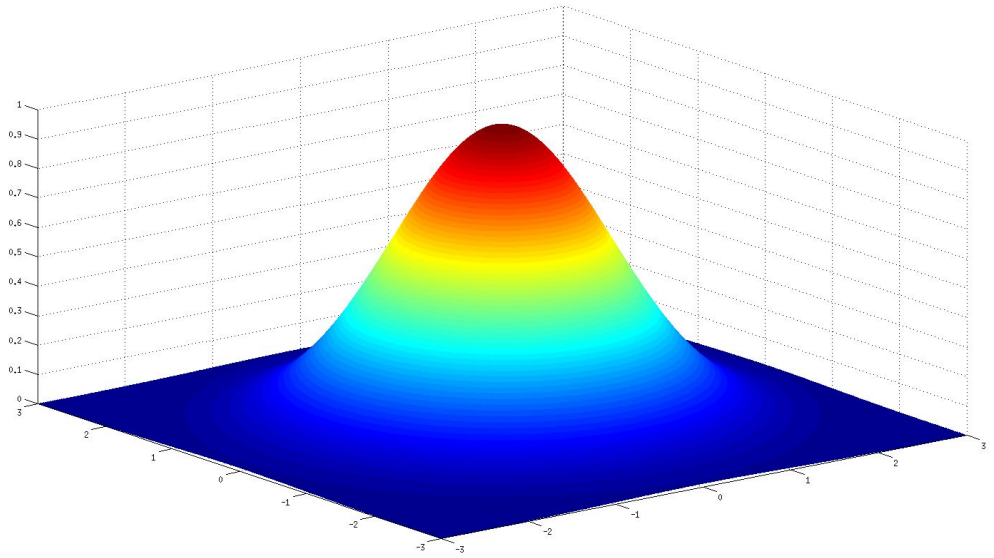
The problem

- Given a series of points (x, y, z) find the value of function in between



Solution

- Use Radial Basis Functions



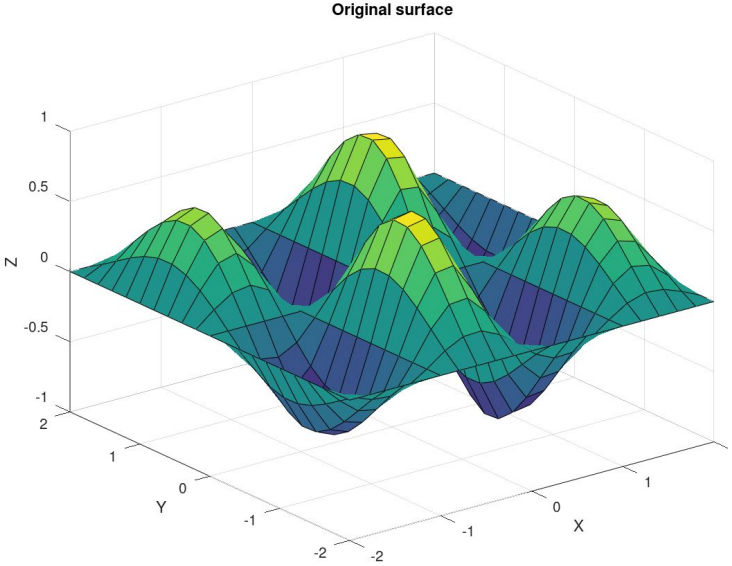
$$f_i = f(\underline{x_i}) = \sum_{j=1}^{K} a_j \Phi\left(\left\|\underline{x_j} - \underline{x_i}\right\|\right) + P(\underline{x_i})$$

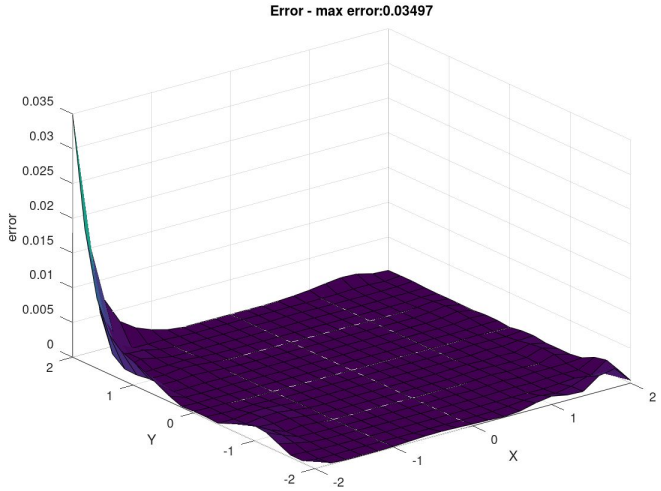
Practical approach

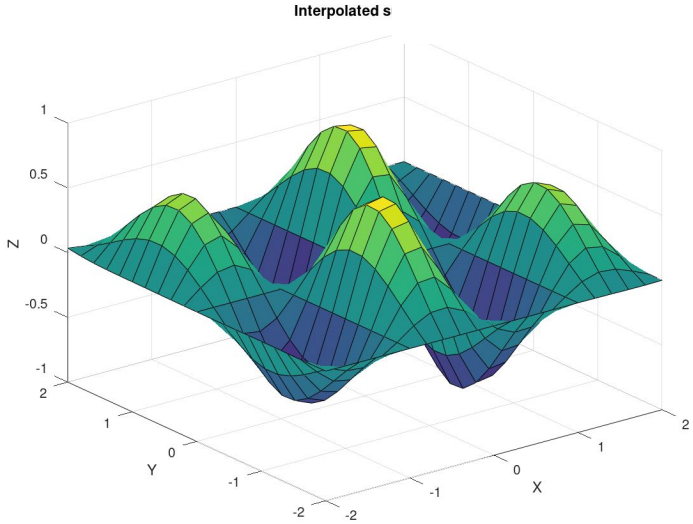
```
function Coeffs = getRbfCoefficients (samplePoints, sampleValues, rbfFunc, beta)
  ndims = size(samplePoints, 2);
  # +1 for constant factor
  ncols = ndims + 1;
  K = length(samplePoints);

  r = zeros(K);

  for i = 1:K
    for j = (i+1):K
      r(i, j) = norm(samplePoints(i, :) - samplePoints(j, :));
      r(j, i) = r(i, j);
    end
  end

  Rbfs = rbf(r, rbfFunc, beta);

  Poly = [ones(K, 1), samplePoints];

  Matrix = [Rbfs, Poly; transpose(Poly), zeros(ncols)];
  y = [sampleValues; zeros(ncols, 1)];

  Coeffs = Matrix \ y;
endfunction
```

```
function [Z] = evaluateRbf (Coeffs, Xknowns, X, rbfFunc, beta)
  K = length(Xknowns);
  N = size(X, 1);

  r = zeros(N, K);

  for i = 1:N
      for j = 1:K
          r(i, j) = norm( X(i, :) - Xknowns(j, :) );
      end
  end

  P = [ones(N, 1), X];
  Rbfs = rbf(r, rbfFunc, beta);

  Z = [Rbfs, P]*Coeffs;
endfunction
```

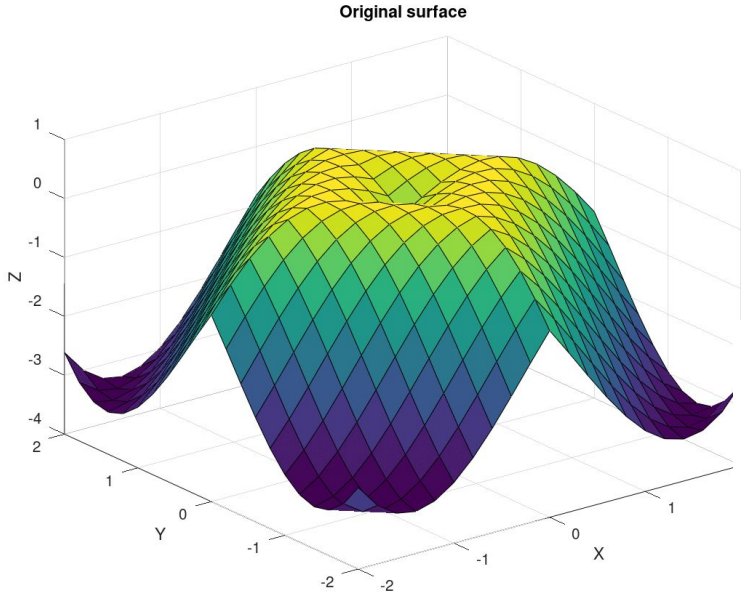
Reproducing lecture results

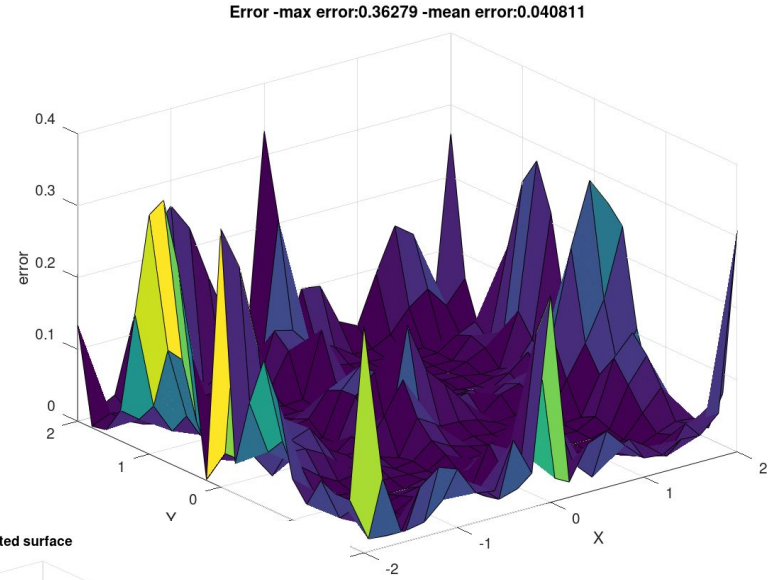


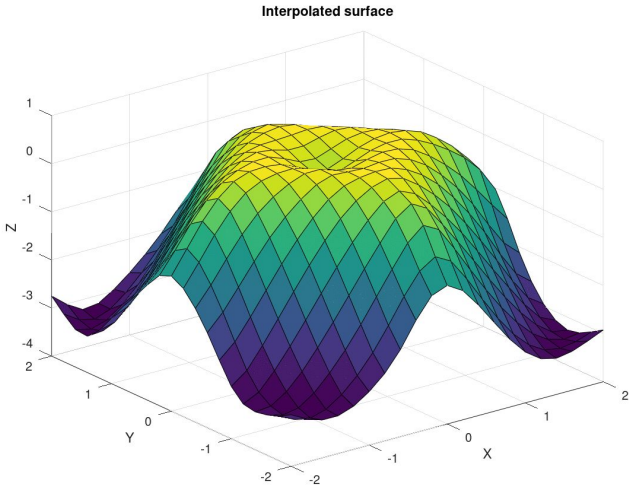




New function



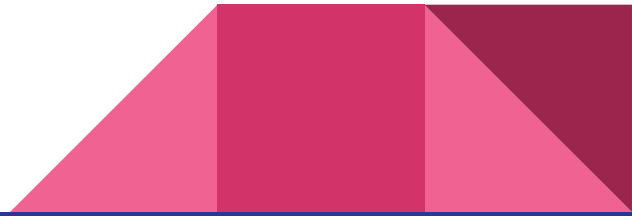




f(x, y) = cos(abs(x) + abs(y)) *(abs(x) + abs(y))

New function approach

- Grid search - 30 different combinations of functions and betas
- Mean and max error measuring
- Measuring time needed for solving the equation and evaluate the interpolation for the rest of the points



Local radial functions

```
function retval = rbfLocal (r, r0)
  retval = zeros(size(r));
  r = r/r0;

  L = (r <= 1);

  #CP_C6
  retval(L) = (1 - r(L)).^8.*(32*r(L).^3 + 25*r(L).^2 + 8*r(L) + 1);
endfunction
```

```
function Coeffs = getRbfCoefficientsLocal (samplePoints, sampleValues, r0)
  ndims = size(samplePoints, 2);
  # +1 for constant factor
  ncols = ndims + 1;
  K = length(samplePoints);

  r = zeros(K);

  for i = 1:K
    for j = (i+1):K
      r(i, j) = norm(samplePoints(i, :) - samplePoints(j, :));
      r(j, i) = r(i, j);
    end
  end

  Rbfs = rbfLocal(r, r0);

  Poly = [ones(K, 1), samplePoints];

  Matrix = sparse([Rbfs, Poly; transpose(Poly), zeros(ncols)]);

  y = [sampleValues; zeros(ncols, 1)];

  Coeffs = Matrix \ y;
endfunction
```

Local radial functions