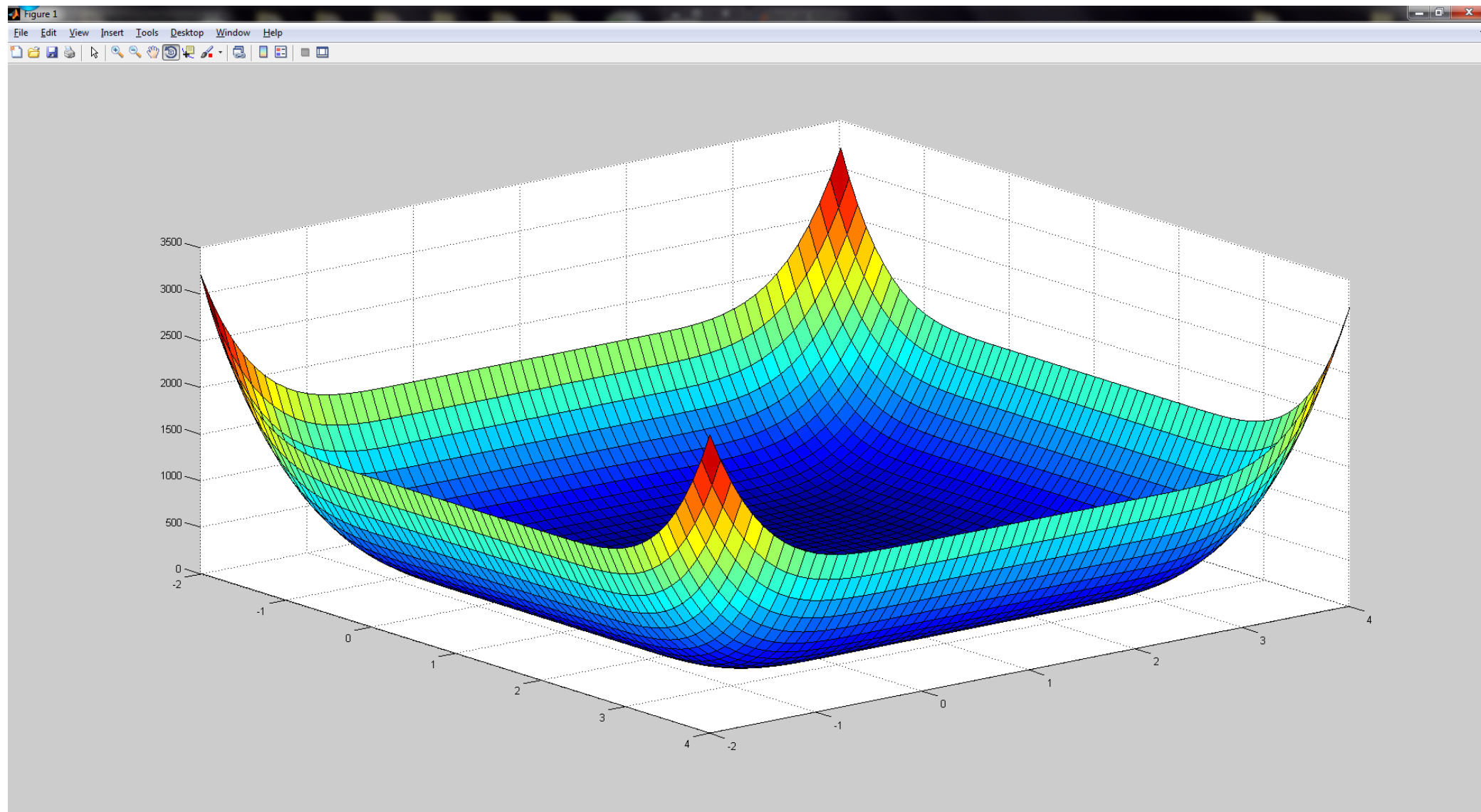
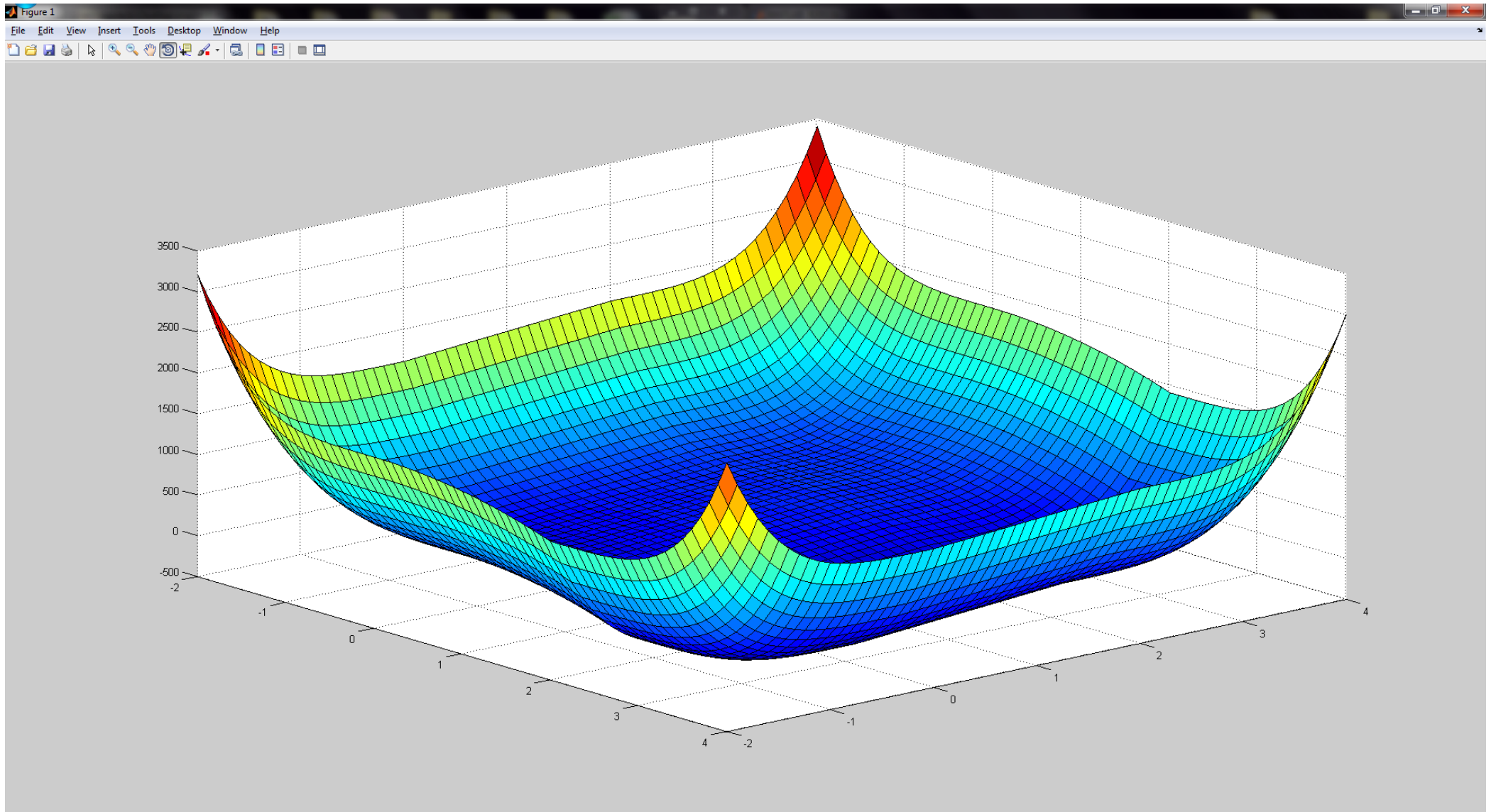


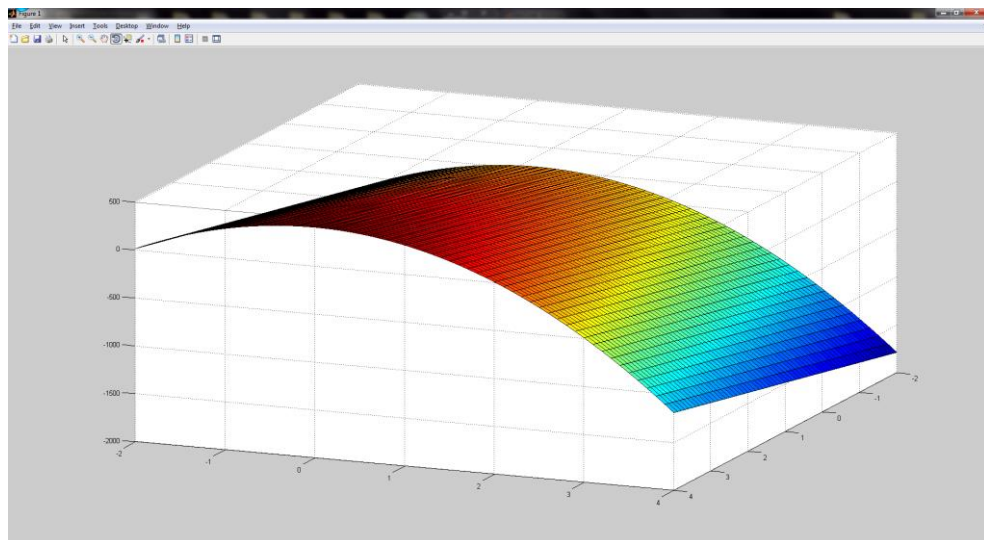
+



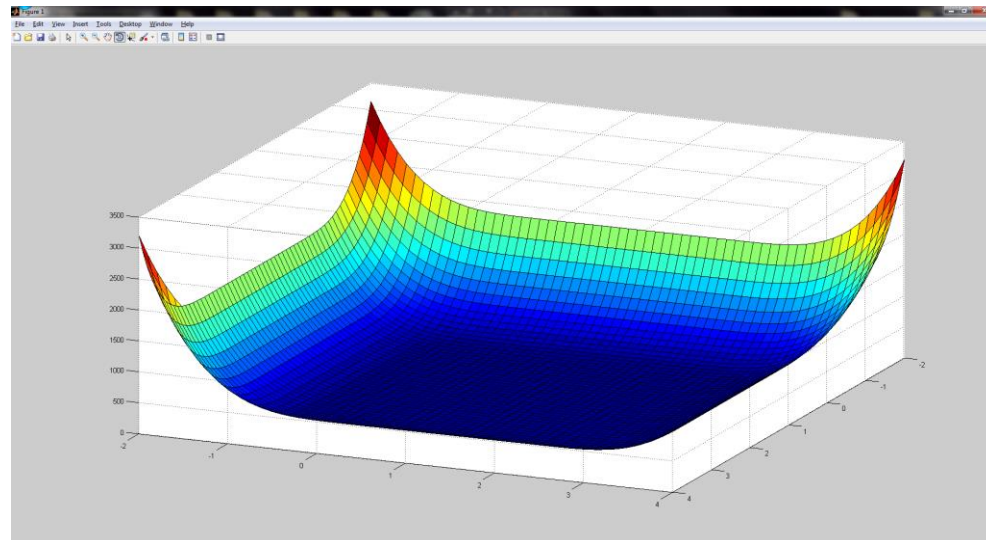
=



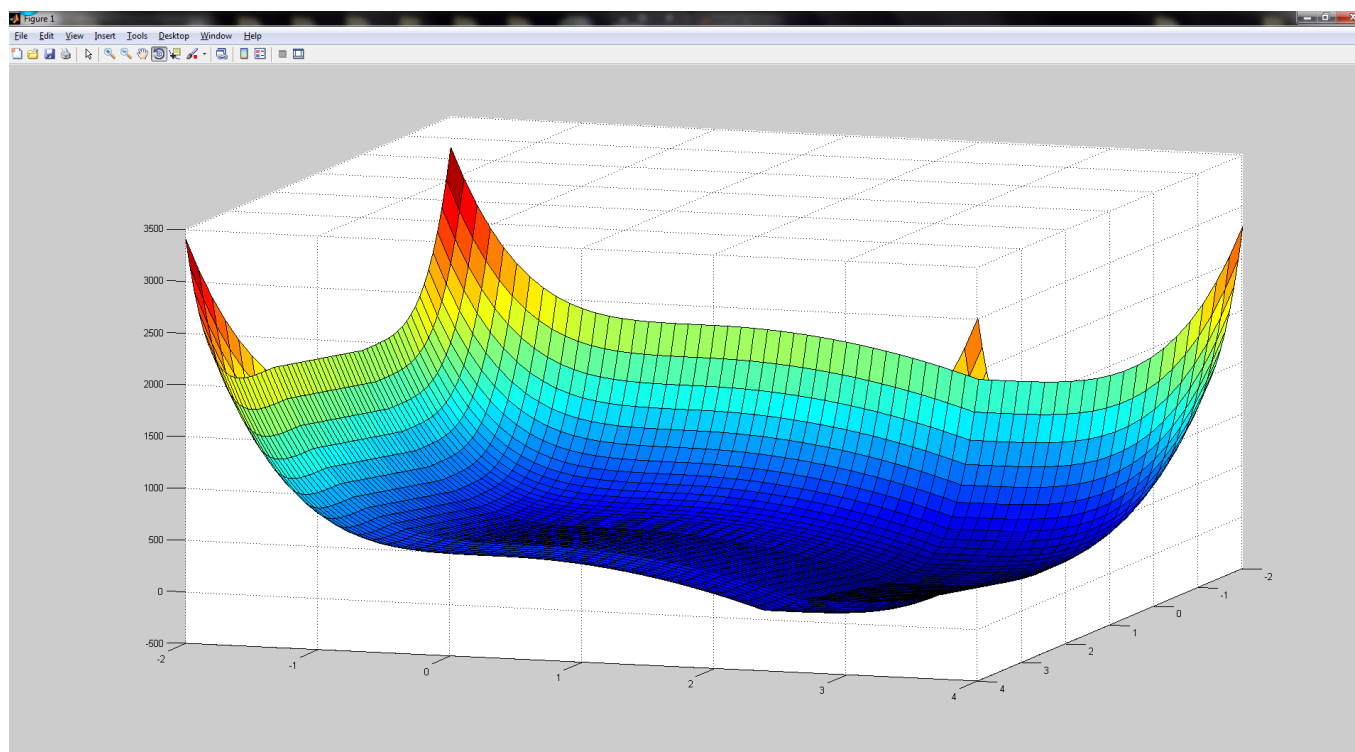
Zrzuty z innej perspektywy:



+



=



Kod programu:

Funkcja obliczająca wartości we wskazanym miejscu

```
%  
% range = [0 2; % x1: min, max % - dopuszczalne zakresy analizowanych argumentow  
%         0 2]; % x2: min, max  
%  
% f = @(x)100 * (x(2) - x(1)^2) + (1 - x(1))^2; % - funkcja celu  
% g = @(x)x^4; % - funkcja "ograniczajaca"  
%  
% n - mnożnik (dzięki niemu możemy zwiększyć kolejne coraz dalsze wartości funkcji kary jeśli badana funkcja jest dosyć  
"stroma")  
%  
function y = compute_goal_function(x, range, f, g, n)  
    tmp = zeros(size(x)); % argumenty przekazywane do funkcji celu  
    p = 0; % wartość o jaką zwiekszamy nasza funkcje (jesli przekroczymy badany zakres)  
  
    for i = 1 : size(x, 1) % dla kazdej wspolrzednej  
        if(x(i) < range(i, 1)) % jesli przekroczono minimum (range(i, 1) to minimum)  
            distance = x(i) - range(i, 1); % distance informuje jak daleko przekroczono zakres na lewo od minimum  
            p = p + n * g(distance);  
            tmp(i) = range(i, 1);  
        else  
            if(x(i) > range(i, 2)) % jesli przekroczono maksimum (range(i, 2) to maksimum)  
                distance = x(i) - range(i, 2); % distance informuje jak daleko przekroczono zakres na prawo od maksimum  
                p = p + n * g(distance);  
                tmp(i) = range(i, 2);  
            else % jesli wartosc mieści sie w badanym przedziale  
                tmp(i) = x(i);  
            end  
        end  
    end  
  
    y = f(tmp) + p;  
    %y = f(x);  
    %y = p;  
end
```

Uruchamiający plik

```
% konfiguracja

range = [0 2; % x1: min, max
         0 2]; % x2: min, max
f = @(x)100 * (x(2) - x(1)^2) + (1 - x(1))^2;
g = @(x)x^4;
n = 100;

t = -2:0.1:4; % okresla wspolrzedne w ktorych rysowane sa punkty

% przyklad uzycia

x=t;
y=t;
z=zeros(length(t));

for i=1 : length(t)
    for j=1 : length(t)
        arguments = [x(j); y(i)];
        z(i, j) = compute_goal_function(arguments, range, f, g, n);
    end
end

surf(x, y, z);
```