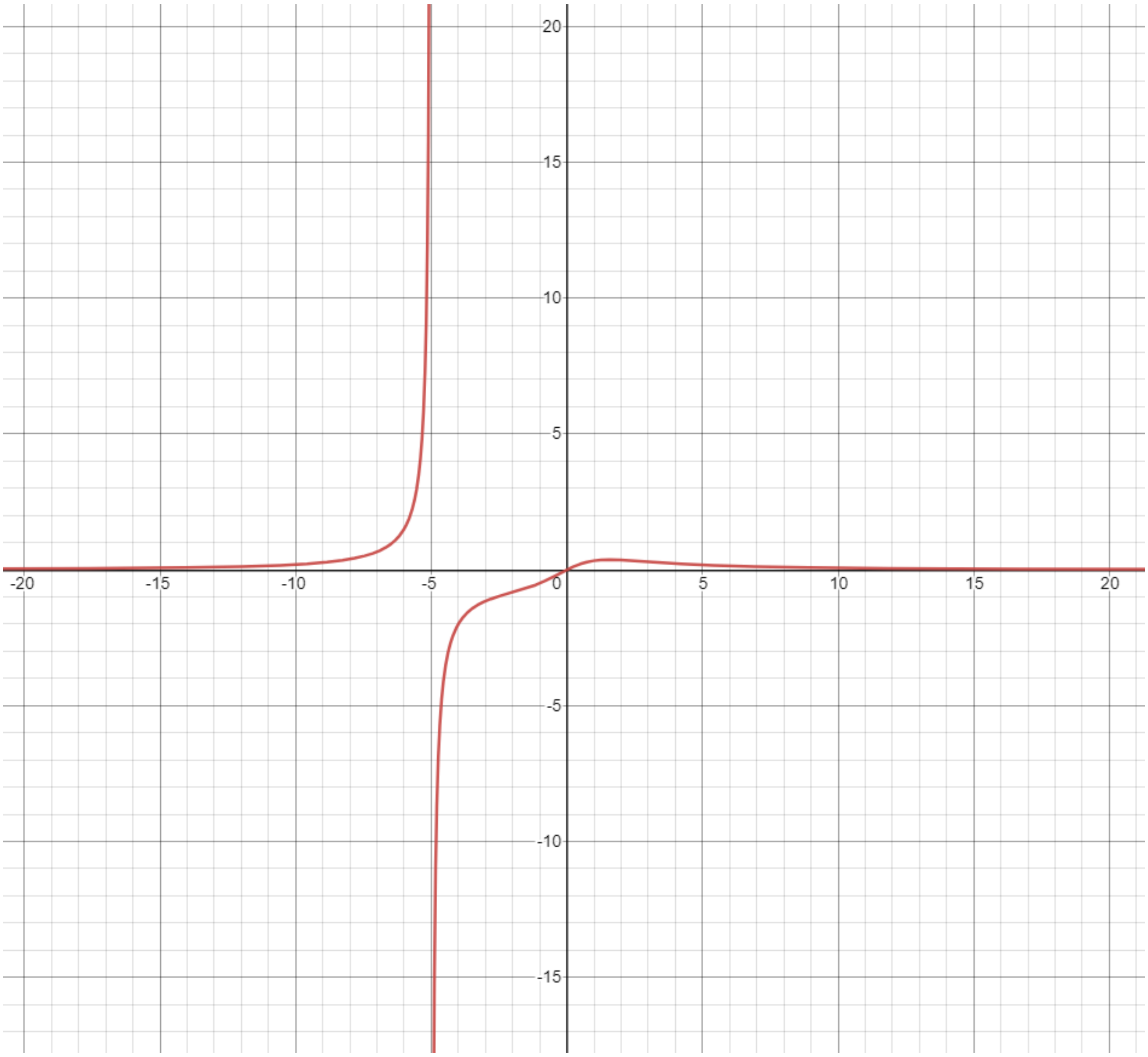
ISC 4220
Continuous Algorithms
Optimization

Computer Lab:

(i) The growth rate, g , of yeast used to produce an antibiotic depends on the concentration of the food c as:

$$g(c) = (2c)/(4+0.8c+(c^2)+(0.2c^3))$$

• Plot the function g(c), and identify a suitable interval which contains the maximum:



Interval where we can find the maximum: (0 , 5)
(asymptotes don't count as either minima or maxima)

- Using the interval determined above as the starting guess, find the maximum using the Golden Section Search method.

Maximum = 1.5679

Proof (code also submitted as Matlab file) :

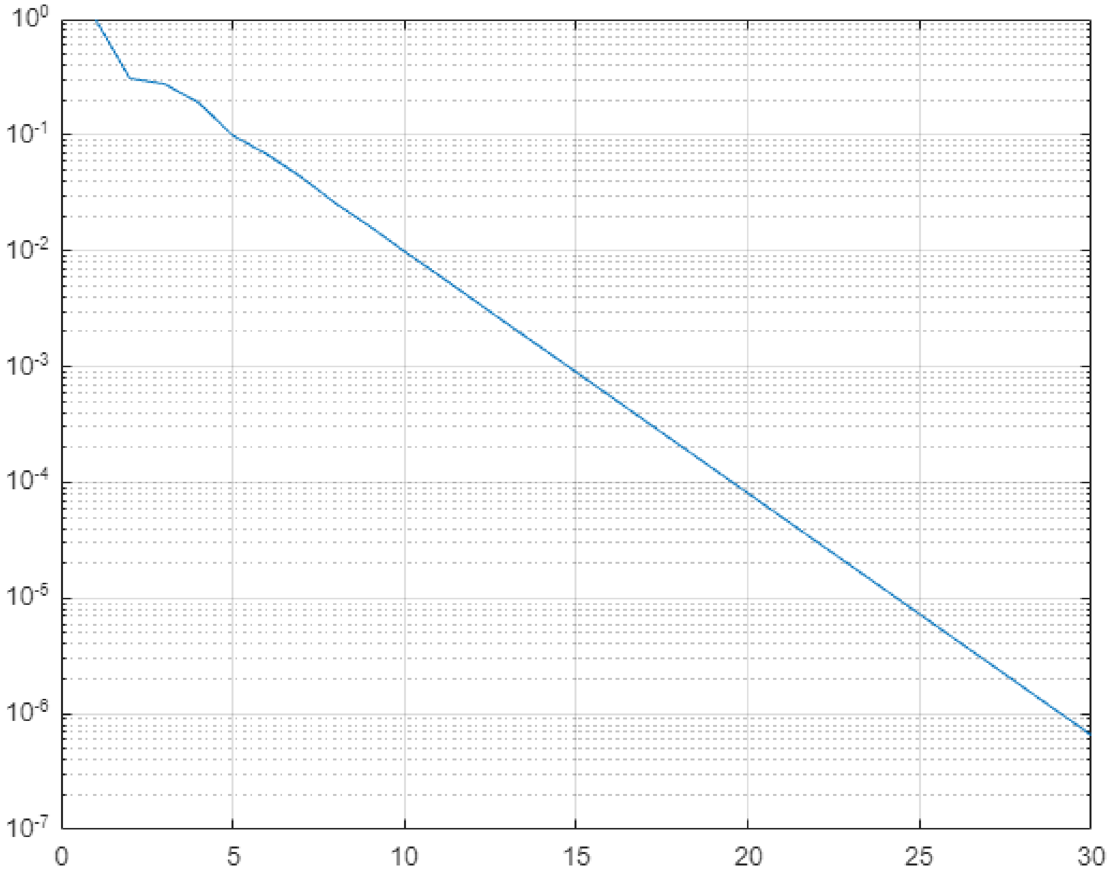
```
% Golden Section Search Method
lowerx = 0;
upperx = 5;
tol= 0.0000001;
MinOrMax = -1;
optima = GoldenSectionSearch(lowerx, upperx, tol, @f, MinOrMax);
disp(optima);
function funct = f(c)
funct = (2*c)/(4+(0.8*c)+(c^2)+(0.2*c^3));
end
function Xopt = GoldenSectionSearch(lowerx, upperx, tol, f, MinOrMax)
Xopt = 0; Error = 1; phi = 1.61803398874989;
goldenR = (phi - 1) * (upperx - lowerx); % calculate golden ratio
if (MinOrMax == -1 )
   %here we are able to change whether we find the minimum (1) or maximum (-1)
         coef = -1;
elseif (MinOrMax == 1)
   coef = 1;
end
x1 = lowerx + goldenR;
x2 = upperx - goldenR;
while (Error > tol)    % runs loop while the error is greater than the stablished tolerance
         if (coef * (f(x1)) < coef * (f(x2)))   %GoldenSectionSearch occurs in if / else statement
       lowerx = x2;
                  x2 = x1;
                  goldenR = (phi - 1) * (upperx - lowerx); %update golden ratio
                  x1 = lowerx + goldenR;
                  Xopt = x1; %update xopt
   else
       upperx = x1;
                  x1 = x2;
                  goldenR = (phi - 1) * (upperx - lowerx); % update golden ratio
                  x2 = upperx - goldenR;
                  Xopt = x2; % update xopt
   end
   Error = (2 - phi) * abs((upperx - lowerx) / Xopt); % update error
end
end
```

- Use a built-in solver like fzero in Matlab or minimize scalar in scipy.optimize to find the “true” maximum c∗ of g(c).

fzero in Matlab isn’t used to find the maximum, it is used to find the root(s) of a function.

The true maximum is: 1.568

- Plot the error i = |xi −x∗|as a function of iteration number i on a semilog plot.

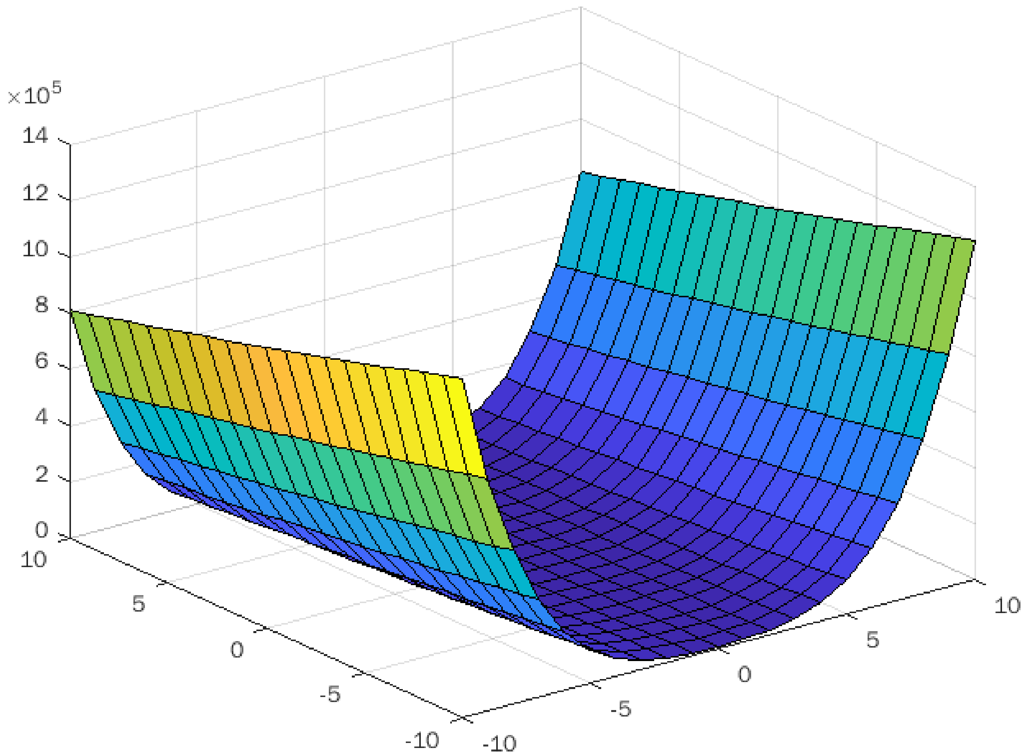


(ii) The Rosenbrock function is often used to test the performance of optimization algorithms, because its global minimum, which occurs at (1,1), lies within a narrow parabolic valley.

f(x, y) = (1-x)^2 + 100(y-x^2)^2

- Plot the function using Matlab's surf function (or equivalent).

```
[x,y] = meshgrid(-10:10,-10:10);
F = ((1-x).^2)+(100*((y-(x.^2)).^2));
surf(x,y,F)
```



- Find the gradient and Hessian of the function:

gradient :

```
[2*x - 400*x*(- x^2 + y) - 2 ; - 200*x^2 + 200*y]
```

Hessian:

```
[1200*x^2 - 400*y + 2, -400*x]
[              -400*x,    200]
```

- Use any multidimensional optimization algorithm to find the minima of the function, from the starting point [−1, 1] . If you do not converge, stop after 50 iterations.

The minimum of the Rosenbrock function is located at [x,y] = [1,1].

(Proof is submitted as matlab code)