# Final Exam

注意：1. 所有題目至少提供程式碼與程式執行結果 (含中間過程)
2. 請使用 MATLAB 程式語言

1. Modify the codes of **genetic algorithm** (using decimal encoding) in the lecture notes to find the maximum of $f(x) = -(x-5)^2 + 4$ in [0, 10].
2. Modify the codes of **ant colony algorithm** in the lecture notes to find the maximum of

$$Ras(x) = 20 + x_1^2 + x_2^2 - 10(\cos 2\pi x_1 + \cos 2\pi x_2)$$

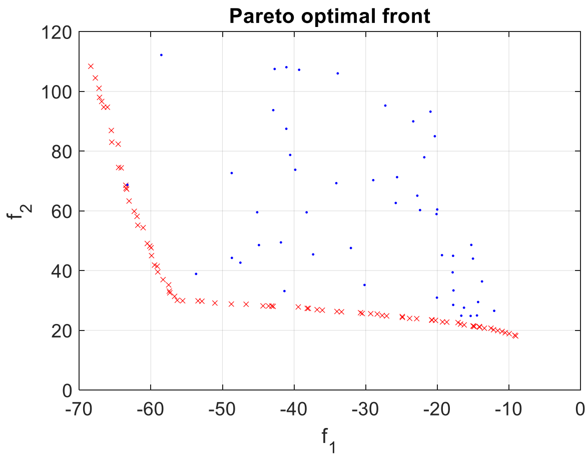
for $-5 \le x_1 \le 5$ and $-5 \le x_2 \le 5$.

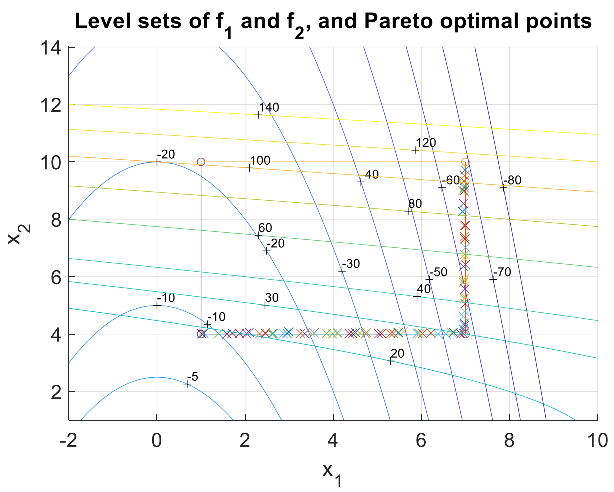
3. Modify the codes demonstrated during the explanation of "Computing the Pareto Front" in the lecture notes to generate the Pareto optimal points and Pareto front on the decision and objective spaces, respectively, for the multiobjective problem:

$$\min \quad \begin{bmatrix} -(x_1^2 + 2x_2) \\ 2x_1 + x_2^2 \end{bmatrix}$$

$$\text{subject to} \quad \begin{array}{l} 1 \le x_1 \le 7 \\ 4 \le x_2 \le 10 \end{array}$$

Both plots should look exactly like the ones shown below.



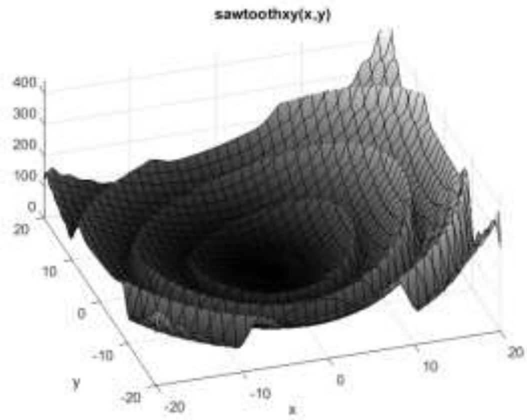




4. Consider the objective function **in polar coordinates** $f(r, \theta) = g(r)h(\theta)$ where

$$g(r) = \left[\sin r - \frac{\sin(2r)}{2} + \frac{\sin(3r)}{3} - \frac{\sin(4r)}{4} + 4\right]\frac{r^2}{r+1}$$

$$h(\theta) = 2 + \cos\theta + \frac{\cos(2\theta - \frac{1}{2})}{2}.$$

(a) Create a surface plot of the function $f$ as shown on the right.




(b) Write MATLAB codes using **GlobalSearch** in MATLAB global optimization toolbox to find the single global minimum of $f$. Use $-100 \le x, y \le 100$ for the bounds. Use the 'sqp' algorithm for fmincon. Use start point [100, -50].

(c) Write MATLAB codes using **MultiStart** in MATLAB global optimization toolbox to find multiple local minimum of $f$. No bounds for $x$ and $y$. Use the fminunc solver. Prepare 20 start points, of which 10 points are randomly generated and the other 10 points center around [100, -50] with suitable statistical variance. List the multiple local minimum obtained.

**Hint:** Incomplete codes for the objective function are provided below for reference.

```
function f = sawtoothxy(x,y)
[t r] = cart2pol(x,y);    % change (x,y) to polar coordinates (t,r)
h = <fill in the codes for h(t)>;
g = <fill in the codes for g(r)>;
f = g.*h;
end
```

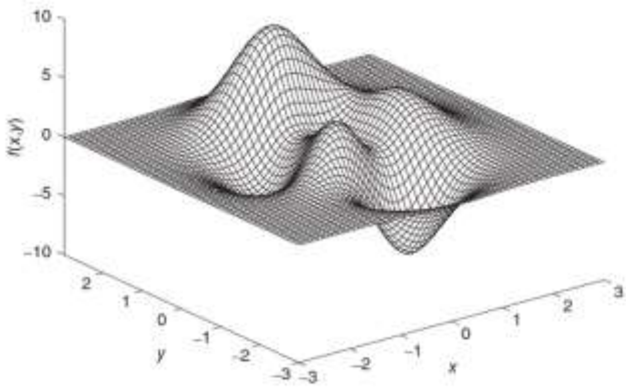
5. Consider the ‘peaks’ function $f$: $R^2 \rightarrow R$ given by

$$f(x,y) = a(1-x^2)e^{-x^2-(y+1)^2} - b\left(\frac{x}{5} - x^3 - y^5\right)e^{-x^2-y^2} - \frac{e^{-(x+1)^2-y^2}}{3}$$

and the following two constraint sets:

$$\Omega_1 = \left\{[x,y]^T : -3 \le x, y \le 3\right\} \text{ and } \Omega_2 = \left\{[x,y]^T : x^2 + y^2 \le 16\right\}.$$

(a) Write MATLAB codes to plot the objective function $f$ with $a$=3 and $b$=10 over the feasible set $\Omega_1$ as shown on the right.




(b) Write MATLAB codes using **ga** in MATLAB global optimization toolbox to find the global maximizer of $f$ with $a$=3 and $b$=10 over the feasible set $\Omega = \Omega_1 \cap \Omega_2$. By tuning the parameters of **optimoptions,** the codes should explore the following aspects:

(i) Compare the results using different ‘Selection Options’: Stochastic uniform sampling vs Remainder stochastic sampling vs Roulette wheel vs Tournament.

(ii) Compare the results using different PopulationSize and ‘Reproduction Options’, i.e., EliteCount, CrossoverFraction, and MutationFcn.

(iii) Select suitable plot functions to visualize the optimization progress.

(iv) Mark the populations at iterations 1, 50, and 100 on the level set plots of the objective function over $\Omega_1$.

6. Consider the scaled version of Rastrigin's function

$$f\left(\frac{\mathbf{x}}{15}\right) = 20 + \left(\frac{x_1}{15}\right)^2 + \left(\frac{x_2}{15}\right)^2 - 10\left(\cos 2\pi\frac{x_1}{15} + \cos 2\pi\frac{x_2}{15}\right)$$

and the constraint set $\Omega = \left\{[x, y]^T : -45 \le x, y \le 45\right\}$.

(a) Write MATLAB codes to plot the objective function *f* over the feasible set Ω.

(b) Write MATLAB codes using **particleswarm** in MATLAB global optimization toolbox to find the global minimizer of *f* over the feasible set $\Omega$. By tuning the parameters of **optimoptions,** the codes should explore the following aspects:

(i) Compare the results using different number of particles.

(ii) After particleswarm finishes its iterations, continue the optimization using @fmincon and see if this will attain better results.

(iii) Use an output function to plot the range that the particles occupy in each dimension. Customize the output plot function as indicated in the lecture slides.