25th February, 2025

STK4400

Mandatory assignment 1 of 1

Submission deadline

Thursday 20th March 2025, 14:30 in Canvas (canvas.uio.no).

Instructions

You can choose between scanning handwritten notes or typing the solution directly on a computer (for instance with LaTeX). The assignment must be submitted as a single PDF file. Scanned pages must be clearly legible. The submission must contain your name, course and assignment number.

It is expected that you give a clear presentation with all necessary explanations. Remember to include all relevant plots and figures. All aids, including collaboration, are allowed, but the submission must be written by you and reflect your understanding of the subject. If we doubt that you have understood the content you have handed in, we may request that you give an oral account.

Application for postponed delivery

If you need to apply for a postponement of the submission deadline due to illness or other reasons, you have to contact the Student Administration at the Department of Mathematics (e-mail: studieinfo@math.uio.no) well before the deadline.

All mandatory assignments in this course must be approved in the same semester, before you are allowed to take the final examination.

Complete guidelines about delivery of mandatory assignments:

uio.no/english/studies/admin/compulsory-activities/mn-math-mandatory.html

GOOD LUCK!

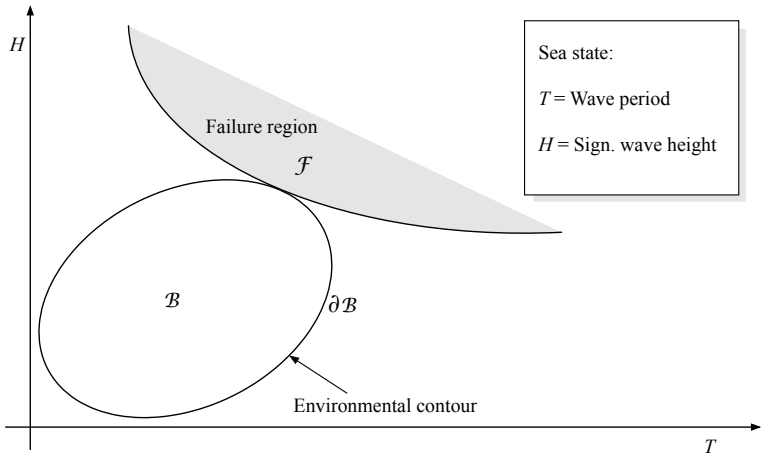


Figure 1: An environmental contour and a potential failure region

Problem 1. In this mandatory assignment we shall study *environmental contours*, and how such contours can be constructed using Monte Carlo simulation. Typically, environmental contours are related to the joint distribution of two *environmental variables*, denoted T and H representing the state of the environment at a given point of time. Here we let:

$$T = \text{The wave period}$$
$$H = \text{The significant wave height}$$

The vector (T, H) describe what is called the *sea state* at a given point of time. Knowledge about the sea state is important in relation to construction of some mechanical structure, e.g., a ship hull. We assume that the structure can be designed in such a way that as long as $(T, H) \in \mathcal{B}$, the structure will not fail. Using the distribution of (T, H), we then want to calculate the *failure probability* of the structure. However, the exact *failure region* of the structure, $\mathcal{F}$, i.e., the set of sea states where the structure fails, is typically *unknown*. Instead we assume that $\mathcal{F}$ belongs to a family of sets $\mathcal{E}$.

Since it is known that the structure does not fail whenever $(T, H) \in \mathcal{B}$, it follows that all sets $\mathcal{F} \in \mathcal{E}$ must be disjoint from $\mathcal{B}$. It is mathematically convenient, however, to allow failure regions to overlap with the boundary of $\mathcal{B}$, denoted $\partial\mathcal{B}$. That is, we assume that $\mathcal{F} \cap \mathcal{B} \subseteq \partial\mathcal{B}$ for all $\mathcal{F} \in \mathcal{E}$. See Figure 1. From this it follows that an upper bound on the failure probability of the structure is given by:

$$P_e(\mathcal{B}, \mathcal{E}) = \sup_{\mathcal{F} \in \mathcal{E}} \{P(T, H) \in \mathcal{F}\}$$

In this context we assume that the set $\mathcal{B}$ as well as all the potential failure regions $\mathcal{F} \in \mathcal{E}$ are all *convex* sets. It can then be shown that:

$$P_e(\mathcal{B}, \mathcal{E}) = \sup_{\theta \in [0,2\pi)} \{P(T\cos(\theta) + H\sin(\theta) \geq B(\mathcal{B}, \theta)\}$$

where the function $B(\mathcal{B}, \cdot)$ is given by:

$$B(\mathcal{B}, \theta) = \sup_{(t,h)\in\mathcal{B}} [t\cos(\theta) + h\sin(\theta)], \quad \theta \in [0, 2\pi)$$

Our main objective is to find a set $\mathcal{B}$ such that:

$$P_e(\mathcal{B}, \mathcal{E}) \leq p_e, \tag{1}$$

where $p_e \in [0, 0.5)$ is a given *target* probability. If $\mathcal{B}$ satisfies (1), we say that $\mathcal{B}$ is a *valid contour set*, or alternatively that the boundary $\partial\mathcal{B}$ is a *valid contour*. In order to find $\mathcal{B}$, we introduce the function $C(\cdot)$, called the p_e-level *percentile function* of the distribution of (T, H), defined by:

$$C(\theta) = \inf_s \{P(T\cos(\theta) + H\sin(\theta) > s) \leq p_e\}, \quad \theta \in [0, 2\pi)$$

We assume that (T, H) is *absolutely continuously distributed*, and hence, it follows that:

$$\begin{aligned} &P(T\cos(\theta) + H\sin(\theta) \geq C(\theta)) \\ &\qquad = P(T\cos(\theta) + H\sin(\theta) > C(\theta)) = p_e, \quad \theta \in [0, 2\pi) \end{aligned}$$

a) Show that if can find a convex set $\mathcal{B}$ such that $B(\mathcal{B}, \theta) \geq C(\theta)$ for all $\theta \in [0, 2\pi)$, then $\mathcal{B}$ is a valid contour set.

Assuming that we can find a convex set $\mathcal{B}$ such that $B(\mathcal{B}, \theta) = C(\theta)$, it can be shown that the set $\mathcal{B}$ is given by:

$$\mathcal{B} = \bigcap_{\theta\in[0,2\pi)} \{(t, h) : t\cos(\theta) + h\sin(\theta) \leq C(\theta)\} \tag{2}$$

We then let $0 \leq \theta_1 < \cdots < \theta_n < 2\pi$, and consider the set:

$$\tilde{\mathcal{B}} = \bigcap_{i=1}^{n} \{(t, h) : t\cos(\theta_i) + h\sin(\theta_i) \leq C(\theta_i)\} \tag{3}$$

b) Explain briefly why $\mathcal{B} \subseteq \tilde{\mathcal{B}}$, and why this implies that $\tilde{\mathcal{B}}$ is also a valid contour set.

The set $\tilde{\mathcal{B}}$ is a convex *polygon* with n corners $(t_1, h_1), \ldots, (t_n, h_n)$, where (t_i, h_i) is determined by the equations:

$$t_i \cos(\theta_i) + h_i \sin(\theta_i) = C(\theta_i)$$
$$t_i \cos(\theta_{i+1}) + h_i \sin(\theta_{i+1}) = C(\theta_{i+1})$$

$i = 1, \ldots, n$, and where we let $\theta_{n+1} = \theta_1$.

c) Show that:

$$t_i = \frac{\sin(\theta_{i+1})C(\theta_i) - \sin(\theta_i)C(\theta_{i+1})}{\sin(\theta_{i+1} - \theta_i)}$$

$$h_i = \frac{-\cos(\theta_{i+1})C(\theta_i) + \cos(\theta_i)C(\theta_{i+1})}{\sin(\theta_{i+1} - \theta_i)}$$

$i = 1, \ldots, n$.

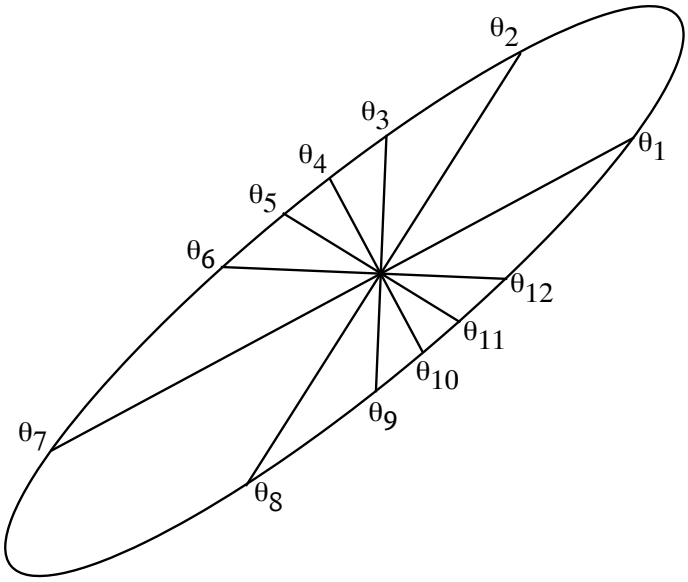


Figure 2: Evenly spaced $\theta_1, \ldots, \theta_n$

When choosing $\theta_1, \ldots, \theta_n \in [0, 2\pi)$, a natural choice would be to let these angles be evenly spaced throughout the interval $[0, 2\pi)$, i.e.:

$$\theta_i = \frac{2\pi i}{n}, \quad i = 0, 1, \ldots, (n-1).$$

However, if the two environmental variables have very different scales, or if the environmental variables are strongly correlated, this may not be the best choice. This issue is illustrated in Figure 2. In this case there is a strong positive correlation between the environment variables. As a result the environmental contour has a skewed oval shape. Using evenly spaced angles the contour points where $C(\theta_i)$ is estimated, will be distributed very unevenly. In particular, in the relatively flat areas around $\theta_3, \ldots, \theta_6$ and $\theta_9, \ldots, \theta_{12}$, the points are close together, while in the curved areas around θ_1, θ_2 and θ_7, θ_8, the points are far apart. This implies e.g., that the tangents through θ_1 and θ_2 will intersect very far away from the true contour. Thus, when approximating $\mathcal{B}$ by the polygon $\tilde{\mathcal{B}}$, the curved areas will be estimated very poorly compared to the flat areas.

One possible solution to this issue is to choose the angles unevenly spaced so that the points where $C(\theta)$ is estimated are more evenly spaced along the contour. This can be accomplished by choosing smaller angles in the curved areas and larger angles in the flat areas. However, this approach requires prior knowledge about the contour in order to choose the best combination of angles. A simpler approach is to *standardize* the environmental variables so that correlations and scale differences are neutralized. More specifically, we assume that:

$$\begin{aligned} \mathrm{E}[T] = \mu_T, \quad \mathrm{Var}[T] = \sigma_T^2, \\ \mathrm{E}[H] = \mu_H, \quad \mathrm{Var}[H] = \sigma_H^2, \end{aligned}$$

and that $\mathrm{Corr}[T, H] = \rho_{T,H} < 1$. We then introduce *standardized* variables U and V, where:

$$\begin{aligned} U &= \frac{T - \mu_T}{\sigma_T} \\ V &= \left[\frac{H - \mu_H}{\sigma_H} - \rho_{T,H} \frac{T - \mu_T}{\sigma_T} \right] / \sqrt{1 - \rho_{T,H}^2}. \end{aligned} \tag{4}$$

d) Show that:

$$\begin{aligned} \mathrm{E}[U] = 0, \quad \mathrm{Var}[U] = 1, \\ \mathrm{E}[V] = 0, \quad \mathrm{Var}[V] = 1, \end{aligned}$$

and that $\mathrm{Corr}[U, V] = 0$.

We then introduce the p_e-level percentile function of the distribution of the standardized (U, V), defined by:

$$C_{\mathrm{STD}}(\theta) = \inf_s \{ P(U \cos(\theta) + V \sin(\theta) > s) \leq p_e \}, \quad \theta \in [0, 2\pi)$$

The corresponding standardized contour set for (U, V) is then given by:

$$\mathcal{B}_{\text{STD}} = \bigcap_{\theta \in [0,2\pi)} \{(u, v) : u\cos(\theta) + v\sin(\theta) \leq C_{\text{STD}}(\theta)\} \tag{5}$$

We then let $0 \leq \theta_1 < \cdots < \theta_n < 2\pi$ be evenly spaced in the interval $[0, 2\pi)$, and consider the set:

$$\tilde{\mathcal{B}}_{\text{STD}} = \bigcap_{i=1}^{n} \{(u, v) : u\cos(\theta_i) + v\sin(\theta_i) \leq C_{\text{STD}}(\theta_i)\} \tag{6}$$

The set $\tilde{\mathcal{B}}_{\text{STD}}$ is a convex polygon with n corners $(u_1, v_1), \ldots, (u_n, v_n)$, where (u_i, v_i) is determined by the equations:

$$u_i\cos(\theta_i) + v_i\sin(\theta_i) = C_{\text{STD}}(\theta_i)$$
$$u_i\cos(\theta_{i+1}) + v_i\sin(\theta_{i+1}) = C_{\text{STD}}(\theta_{i+1})$$

$i = 1, \ldots, n$, and where we let $\theta_{n+1} = \theta_1$.

Finally, the contour $\mathcal{B}$ is approximated by a convex polygon $\tilde{\mathcal{B}}$ with n corners $(t_1, h_1), \ldots, (t_n, h_n)$ determined by:

$$t_i = \sigma_T u_i + \mu_T,$$
$$h_i = \sigma_H \left[\rho_{T,H} u_i + \sqrt{1 - \rho_{T,H}^2}\, v_i\right] + \mu_H$$

for $i = 1, \ldots, n$. In cases where μ_T, σ_T, μ_H, σ_H and $\rho_{T,H}$ are unknown, these quantities can easily be estimated by Monte Carlo simulation.

Since there is a simple linear relation between the original environmental variables (T, H) and the standardized variables (U, V), it can be shown that $\mathcal{B}$ and $\mathcal{B}_{\text{STD}}$ have the same probabilistic properties. That is, $\mathcal{B}$ is a valid contour set for (T, H) if and only if $\mathcal{B}_{\text{STD}}$ is a valid contour set for (U, V).

In the remaining part of this assignment we consider the problem of estimating environmental contours using Monte Carlo simulation. We start out by assuming that we have generated independent vectors:

$$(T_1, H_1), \ldots, (T_N, H_N)$$

with the same distribution as (T, H). We then let $\theta \in [0, 2\pi)$ and introduce:

$$S_j(\theta) = T_j\cos(\theta) + H_j\sin(\theta), \quad j = 1, \ldots, N.$$

e) Explain briefly how $C(\theta)$ can be estimated using the generated data:

$$S_1(\theta), \ldots, S_N(\theta).$$

The target probability, p_e, is typically *very small*, which implies that the environmental contour $\partial\mathcal{B}$ is located far out in the *tail area* of the joint distribution. As a result, only a tiny fraction of the sampled data, $(T_1, H_1), \ldots, (T_N, H_N)$ will fall in this area. Thus, unless N is extremely large, the estimate of $C(\theta)$ will be very unstable. Fortunately, this issue can be handled by using *importance sampling*. More specifically, we assume that it is possible to sample $(T_1, H_1), \ldots, (T_N, H_N)$ from the conditional distribution of (T, H) given that $(T, H) \notin \mathcal{D}$, where $\mathcal{D} \subset \mathcal{B}$, and where:

$$P((T, H) \notin \mathcal{D}) = p_{\mathcal{D}}$$

is *known*. We then introduce the *adjusted target probability*:

$$p'_e = P(T\cos(\theta) + H\sin(\theta) > C(\theta) | (T, H) \notin \mathcal{D})$$

f) Show that the adjusted target probability is given by:

$$p'_e = \frac{p_e}{p_{\mathcal{D}}}$$

and explain how $C(\theta)$ can be estimated using the data:

$$S_1(\theta), \ldots, S_N(\theta).$$

assuming that $(T_1, H_1), \ldots, (T_N, H_N)$ are sampled from the conditional distribution of (T, H) given that $(T, H) \notin \mathcal{D}$.

We then assume that X and Y are independent standard normally distributed. Sampling $(T_1, H_1), \ldots, (T_N, H_N)$ from the same distribution as (T, H) can then be done by sampling $(X_1, Y_1), \ldots, (X_N, Y_N)$ from the same distribution as (X, Y), and let:

$$(T_j, H_j) = \Psi^{-1}(X_j, Y_j), \quad j = 1, \ldots, N,$$

where Ψ denotes the *Rosenblatt transformation*.

Furthermore, let $\mathcal{O}$ denote a circle centered in the origin with radius r_0, where r_0 is chosen so that:

$$\mathcal{D} = \Psi^{-1}(\mathcal{O}) = \{(t, h) = \Psi^{-1}(x, y) : (x, y) \in \mathcal{O}\} \subset \mathcal{B}. \tag{7}$$

Then $(T_1, H_1), \ldots, (T_N, H_N)$ can be sampled from conditional distribution of (T, H) given that $(T, H) \notin \mathcal{D}$, by sampling $(X_1, Y_1), \ldots, (X_N, Y_N)$ from the conditional distribution of (X, Y) given that $(X, Y) \notin \mathcal{O}$.

More specifically, we let:

$$r_0 = \alpha\, r,$$

where r is the $(1 - p_e)$-percentile of the standard normal distribution, and α is a non-negative factor referred to as the *radius factor*. We observe that if $\alpha = 0$, then $\mathcal{O} = \{(0, 0)\}$. This implies that the set $\mathcal{D}$ contains only a single point: $\Psi^{-1}(0, 0)$. In this case $p_{\mathcal{D}} = 1$, and hence, $p'_e = p_e$. Thus, in this case the conditional distribution of (T, H) given that $(T, H) \notin \mathcal{D}$ is essentially the same as the unconditional distribution of (T, H). As the radius factor α increases, the the effect of the importance sampling becomes more noticeable since a higher fraction of the sampled data will be located in the tail area of the joint distribution. However, if α is too large, the condition (7) will fail to hold true. Experience has shown that $\alpha = 0.95$ is a good choice.

In the simulation study the joint probability distribution for (T, H) is given by:

$$f_{T,H}(t, h) = f_H(h) f_{T|H}(t|h)$$

where a three-parameter Weibull distribution is used for H, and a lognormal conditional distribution is used for T. The Weibull distribution is parameterized by a *scale parameter*, α, a *shape parameter*, β, and a *location parameter*, γ, and is given by:

$$f_H(h) = \frac{\beta}{\alpha}\left(\frac{h-\gamma}{\alpha}\right)^{\beta-1} e^{-[(h-\gamma)/\alpha]^{\beta}}, \quad h \geq \gamma.$$

The lognormal distribution has two parameters, the *log-mean* μ and the *log-standard deviation* σ and is expressed as:

$$f_{T|H}(t|h) = \frac{1}{t\sqrt{2\pi}} e^{-[(\ln(t)-\mu)^2/(2\sigma^2)]}, \quad t \geq 0,$$

The dependence between H and T is modelled by letting the parameters μ and σ be expressed in terms of H as follows:

$$\mu = E[\ln(T)|H = h] = a_1 + a_2 h^{a_3},$$
$$\sigma = SD[\ln(T)|H = h] = b_1 + b_2 e^{b_3 h}.$$

The parameters α, β, γ, a_1, a_2, a_3 and b_1, b_2, b_3 are estimated based on wave data from the relevant location.

The Monte Carlo simulation procedure is implemented in the form of a Python program. The program consists of three files:

- `c_contour.py`
- `util.py`
- `contourtest_01.py`

The file `c_contour.py` contains most of the code needed to run the Monte Carlo simulations and estimate environmental contours, while the file `util.py` contains various utility functions. The file `contourtest_01.py`, contains the main program, and includes a number of input settings listed and explained in Figure 3. Note that the input settings are divided into two groups. The *upper group* of input settings controls the numerical properties of the simulations, while the *lower group* of input settings controls the various types of output the program produces. The last input variable, denoted `save_plots` controls whether or not the plots should be saved as pdf-files.

IMPORTANT: When answering the points below, make sure you include *all* the saved plots. Before letting `save_plots = True`, however, a folder called `contourtest01` must be created in the directory where the program files are located. The resulting pdf-files will be stored in this folder.

```
################################################
#             -- Input settings --             #
################################################
radiusFactor = 0.95         # <-------- Choose the radius factor (< 1) here
numSimulations = 25000      # <-------- Choose the number of simulations here
numTangents = 360           # <-------- Choose the number of tangents here
doStandardize = True        # <-------- Choose if the data should be standardized here
smoothWidth = 6             # <-------- Choose smoothness width here (0 = no smoothing)
returnPeriod = 10           # <-------- Choose the return period here (in years)
tr = wShetlandTotalSea      # <-------- Choose the environment contour scenario here

printMoments = False        # <-------- Choose if the moments should be printed out
printC = False              # <-------- Choose if the c-function should be printed out
printContour = False        # <-------- Choose if the contour points should be printed out
plotC = False               # <-------- Choose if c(theta) plot should be created
plotDC = False              # <-------- Choose if the c'(theta) plot should be created
plotDC2 = False             # <-------- Choose if the c''(theta) plot should be created
plotCDC2 = False            # <-------- Choose if the c(theta) + c''(theta) plot should be created
plotScatter = False         # <-------- Choose if a scatter plot should be created
plotContour = True          # <-------- Choose if a contour curve plot should be created
save_plots = False          # <-------- Choose if the plot(s) should be saved here
```

Figure 3: Input settings for the Python program

In the simulation study we focus on *one particular scenario* based on wave data from West Shetland. In the Python code, this scenario is referred to as `wShetlandTotalSea`. In order to choose this particular scenario, make sure that the input variable `tr` is always set to `wShetlandTotalSea`.

g) In this point we use the following input settings for the *upper group*:

- `radiusFactor = 0.95`
- `numSimulations = 25000`
- `numTangents = 360`
- `doStandardize = True`
- `smoothWidth = 0`

The lower group of input settings should all be set to `False` except:

- `plotContour = True`
- `save_plots = True`

Run the program with `returnPeriod = 10`, `returnPeriod = 25` and `returnPeriod = 50`, and compare the resulting environmental contours.

In the remaining points we let `returnPeriod = 10`.

h) In this point we use the following input settings for the *upper group*:

- `numSimulations = 25000`
- `numTangents = 360`
- `doStandardize = True`
- `smoothWidth = 0`

The lower group of input settings should all be set to `False` except:

- `plotScatter = True`
- `save_plots = True`

Run the program with `radiusFactor = 0.00`, `radiusFactor = 0.50`, `radiusFactor = 0.95` and `radiusFactor = 0.99`, and compare the resulting environmental contours. Describe how the different radius factor values affect the shape of the environmental contour.

In the remaining points we let `radiusFactor = 0.95`.

i) In this point we use the following input settings for the *upper group*:

- `numSimulations = 25000`

- `doStandardize = True`
- `smoothWidth = 0`

The lower group of input settings should all be set to `False` except:

- `plotContour = True`
- `save_plots = True`

The variable `numTangents` represents the number of θ-values used in the approximation of the contour, and corresponds to the variable n above. This is the same as the number of sides in the polygon. Run the program with `numTangents = 36`, `numTangents = 72` and `numTangents = 360`, and compare the resulting environmental contours. Describe how the number of tangents affect the shape of the environmental contour.

j) In this point we use the following input settings for the *upper group*:

- `radiusFactor = 0.95`
- `numSimulations = 25000`
- `numTangents = 360`
- `doStandardize = True`
- `smoothWidth = 6`

The lower group of input settings should all be set to `False` except:

- `plotC = True`
- `plotDC = True`
- `plotDC2 = True`
- `plotCDC2 = True`
- `plotContour = True`
- `save_plots = True`

We now focus on the integer input variable `smoothWidth`. This variable indicates that the estimate of the C-function is smoothened using a moving weighted average. If `smoothWidth = 0`, as it has been in the previous points, no smoothing is applied. In this point, however, we let `smoothWidth = 6`. As a result the moving weighted average window has length $6 \cdot 2 + 1 = 13$.

In order to study the effect of the smoothing, we include a plot of the estimated standardized C-function (which is included since `plotC = True`), a plot of the estimated derivative of the standardized C-function (which is included since `plotDC = True`), a plot of the estimated second derivative of the standardized C-function (which is included since `plotDC2 = True`), and a plot of the sum of the estimated standardized C-function and its estimated second derivative (which is included since `plotCDC2 = True`).

If the environmental contour $\partial\mathcal{B}$ is convex (as it should be) the sum of the estimated standardized C-function and its estimated second derivative should be positive for all θ. Due to Monte Carlo noise, however, this is not always the case. As a result the environmental contour set will have characteristic irregularities along its border.

By smoothening the estimated standardized C-function, this issue can be eliminated. If the program is run with a positive `smoothWidth`, the estimated curves as well as the environmental contour is calculated twice, first without smoothing, and then with smoothing. The final environmental contour plot includes both the original contour without smoothing, as well as the smoothed contour, making it easy to compare the two. Run the program with the above settings, and describe the results.