

Μέθοδοι Βελτιστοποίησης

2^η Εργασία

Θέμα στις Μεθόδους Gradient-based Optimization

Ονοματεπώνυμο: Μιχαήλ Κουσκούνης

Αριθμός μητρώου: mc21052

Κατεύθυνση: Ενεργειακή

Πίνακας Περιεχομένων

1 Εισαγωγή.....	3
2 Διατύπωση της συνεχούς συζυγής μεθόδου	4
2.1 Το primal πρόβλημα	4
2.2 Η συνεχούς συζυγής μέθοδος	4
2.3 Διαχείριση περιορισμού κάθετου τοίχου	10
3 Μέθοδος πεπερασμένων διαφορών (FD).....	11
4 Μέθοδος συζυγών μεταβλητών (CV)	11
5 Αριθμητικά αποτελέσματα	12
5.1 Επιλογή βήματος Runge-Kutta.....	12
5.2 Σύγκριση παραγώγων ευαισθησίας για διάφορες τιμές του ε	13
5.3 Βελτιστοποίηση με adjoint χωρίς το περιορισμό του κάθετου τοίχου	14
5.4 Βελτιστοποίηση με FD χωρίς το περιορισμό του κάθετου τοίχου	15
5.5 Βελτιστοποίηση με adjoint με το περιορισμό του κάθετου τοίχου	16
6 Παράρτημα κώδικα.....	18
6.1 Κώδικας για την FD	18
6.2 Κώδικας adjoint.....	21
6.3 Κώδικας CV.....	26

Πίνακας Διαγραμμάτων

Διάγραμμα 1: Τροχιές για διάφορα χρονικά βήματα.....	12
Διάγραμμα 2: Τροχιά μάζας χωρίς τον κάθετο τοίχο με μέθοδο adjoint.	14
Διάγραμμα 3: Σύγκλιση χρόνου T συναρτήσει κύκλων βελτιστοποίησης με μέθοδο adjoint.	15
Διάγραμμα 4: Τροχιά μάζας με μέθοδο FD.....	15
Διάγραμμα 5: Σύγκλιση χρόνου T συναρτήσει κύκλων βελτιστοποίησης με μέθοδο FD.....	16
Διάγραμμα 6: Τροχιά μάζας όταν υπάρχει ενδιάμεσος τοίχος με μέθοδο adjoint.....	17
Διάγραμμα 7: Σύγκλιση χρόνου T συναρτήσει κύκλων βελτιστοποίησης με μέθοδο adjoint.	17

Πίνακας Πινάκων

Πίνακας 1: Τιμές παραγώγων με μέθοδο FD.	13
Πίνακας 2: Τιμές παραγώγων με μέθοδο CV.....	13
Πίνακας 3: Παράγωγοι από τη συνεχή συζυγή μέθοδο.....	13

1 Εισαγωγή

Στα πλαίσια της 2^{ης} εργασίας του μαθήματος «Μέθοδοι Βελτιστοποίησης» ζητείται η ελαχιστοποίηση του χρόνου της βολής T μίας σφαίρας η οποία περνάει μέσα από το άνοιγμα ενός τοίχου. Για την επίλυσή του θα χρησιμοποιηθεί η συνεχής συζυγής μέθοδος της οποίας οι παράγωγοι ευαισθησίας θα συγκριθούν με αυτές που προκύπτουν από την μέθοδο των πεπερασμένων διαφορών και των μιγαδικών μεταβλητών.

2 Διατύπωση της συνεχούς συζυγής μεθόδου

2.1 Το primal πρόβλημα

Πριν γίνει η ανάλυση της adjoint, πρέπει να παρουσιαστεί το πρωτεύον πρόβλημα. Το πρωτεύον πρόβλημα απαρτίζεται από τις εξισώσεις (I) και (II) οι οποίες λαμβάνουν υπόψιν και την αντίσταση του αέρα. Ο όρος m είναι η συνολική μάζα της σφαίρας, ρ είναι η πυκνότητα του αέρα, C_d ο συντελεστής αντίστασης του αέρα και A το εμβαδό της εμπρόσθιας όψης της σφαίρας.

$$\frac{d^2x}{dt^2} = -\frac{1}{2m}\rho C_d A \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} \frac{dx}{dt} \quad (I)$$

$$\frac{d^2y}{dt^2} = -g - \frac{1}{2m}\rho C_d A \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} \frac{dy}{dt} \quad (II)$$

Η επίλυση των παραπάνω εξισώσεων γίνεται με τη μέθοδο Runge-Kutta 4^{ης} τάξης. Επειδή στη συνέχεια θα εμφανιστεί πολλές φορές ο υπόριζος όρος, ορίζεται ότι: $|V| = \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}$.

2.2 Η συνεχής συζυγής μέθοδος

Στο παρόν πρόβλημα, οι μεταβλητές σχεδιασμού είναι η οριζόντια ταχύτητα στην αρχική θέση $b_1 = u_0$, η κάθετη ταχύτητα στην αρχική θέση $b_2 = v_0$ και η μάζα του έρματος $b_3 = E$. Επειδή το E ως μεταβλητή σχεδιασμού αλλάζει σε κάθε επανάληψη, αλλάζουν και οι διαστάσεις της σφαίρας. Πιο συγκεκριμένα, για τον όγκο της σφαίρας ισχύει ότι:

$$V = \frac{4}{3}\pi R^3 = \frac{0.5}{\rho_1} + \frac{E}{\rho_2} \Rightarrow R = \sqrt[3]{\frac{3}{4\pi} \left(\frac{0.5}{\rho_1} + \frac{E}{\rho_2} \right)}$$

Επομένως, η μεταβλητή A εξαρτάται από το E σύμφωνα με την εξίσωση (III).

$$A = \pi \left(\frac{3}{4\pi} \right)^{2/3} \left(\frac{0.5}{\rho_1} + \frac{E}{\rho_2} \right)^{2/3} \quad (III)$$

Η επαυξημένη αντικειμενική συνάρτηση που θα χρησιμοποιηθεί για την adjoint είναι η εξής:

$$F_{aug} = T + \int_0^T \Psi \left(\frac{d^2x}{dt^2} + \frac{1}{2m} \rho C_d A |V| \frac{dx}{dt} \right) dt + \Phi \int_0^T \left(\frac{d^2y}{dt^2} + g + \frac{1}{2m} \rho C_d A |V| \frac{dy}{dt} \right) dt + M(x(T) - x_{tar})^2 + Qy(T) \quad (IV)$$

Στην επαυξημένη, εκτός από τις εξισώσεις (I) και (II) έχουν προστεθεί και 2 επιπλέον όροι, ένας για τον άξονα x και ένας για τον άξονα y. Επειδή η μέθοδος δεν μπορεί να διαχειριστεί και τους 2 περιορισμούς ταυτόχρονα, γίνεται χρήση και της μεθόδου ALM. Ο περιορισμός για χρόνο T η σφαίρα να βρίσκεται στο έδαφος ελέγχεται από την adjoint και ο περιορισμός για χρόνο T η σφαίρα να βρίσκεται στο $x(T)=x_{tar}$ ελέγχεται από την ALM. Επομένως υπάρχουν 2 λούπες, μια εσωτερική η οποία βελτιστοποιεί την βολή για συγκεκριμένο συντελεστή ποινής M, και μια εξωτερική η οποία αυξάνει αυτό τον συντελεστή. Σε αυτό το σημείο αγνοούμε την ύπαρξη του τοίχου στο κέντρο η οποία προσθέτει άλλους 2 περιορισμούς.

Στη συνέχεια, η F_{aug} θα παραγωγιστεί ως προς b_n . Ο κάθε όρος της εξίσωσης (IV) θα αναλυθεί ξεχωριστά. Αγνοώντας τον 1^ο όρο της (IV) καθώς είναι εύκολη η παραγωγή του, για τον 2^ο όρο έχουμε ότι:

$$\frac{\delta}{\delta b_n} \int_0^T \Psi \left(\frac{d^2x}{dt^2} + \frac{1}{2m} \rho C_d A |V| \frac{dx}{dt} \right) dt = \int_0^T \Psi \frac{\partial}{\partial b_n} \left(\frac{d^2x}{dt^2} + \frac{1}{2m} \rho C_d A |V| \frac{dx}{dt} \right) dt + \Psi \left(\frac{d^2x}{dt^2} + \frac{1}{2m} \rho C_d A |V| \frac{dx}{dt} \right) \Big|_T \frac{\delta T}{\delta b_n} - \Psi \left(\frac{d^2x}{dt^2} + \frac{1}{2m} \rho C_d A |V| \frac{dx}{dt} \right) \Big|_0 \frac{\delta(0)}{\delta b_n} \quad (V)$$

Οι σταθεροί όροι που προκύπτουν λόγω του κανόνα του Leibniz μηδενίζονται καθώς η εξίσωση (I) ισχύει πάντα. Άρα, παραμένει μόνο το ολοκλήρωμα, το οποίο αναπτύσσεται ως εξής:

$$\begin{aligned} \int_0^T \Psi \frac{\partial}{\partial b_n} \left(\frac{d^2x}{dt^2} + \frac{1}{2m} \rho C_d A |V| \frac{dx}{dt} \right) dt &= \int_0^T \Psi \frac{\partial}{\partial b_n} \left(\frac{d^2x}{dt^2} \right) dt + \\ \int_0^T \Psi \frac{\partial}{\partial b_n} \left(\frac{1}{2m} \right) \rho C_d A |V| \frac{dx}{dt} dt &+ \int_0^T \Psi \frac{1}{2m} \rho C_d \frac{\partial A}{\partial b_n} |V| \frac{dx}{dt} dt + \\ \int_0^T \Psi \frac{1}{2m} \rho C_d A \frac{\partial |V|}{\partial b_n} \frac{dx}{dt} dt &+ \int_0^T \Psi \frac{1}{2m} \rho C_d A |V| \frac{\partial}{\partial b_n} \left(\frac{dx}{dt} \right) dt \end{aligned} \quad (VI)$$

Ο κάθε όρος της εξίσωσης (VI) θα αναλυθεί επίσης ξεχωριστά. Για τον 1^ο όρο έχουμε ότι:

$$\int_0^T \Psi \frac{\partial}{\partial b_n} \left(\frac{d^2x}{dt^2} \right) dt = \Psi(T) \frac{\partial}{\partial b_n} \left(\frac{dx}{dt} \right) \Big|_T - \Psi(0) \delta_1^n - \frac{d\Psi}{dt} \Big|_T \frac{\partial x(T)}{\partial b_n} + \int_0^T \frac{\partial x}{\partial b_n} \left(\frac{d^2\Psi}{dt^2} \right) dt \quad (VII)$$

Για τον 2^ο όρο έχουμε ότι:

$$\int_0^T \Psi \frac{\partial}{\partial b_n} \left(\frac{1}{2m} \right) \rho C_d A |V| \frac{dx}{dt} dt = - \int_0^T \Psi \frac{1}{2m^2} \delta_3^n \rho C_d A |V| \frac{dx}{dt} dt \quad (VIII)$$

Για τον 3^ο όρο ισχύει ότι:

$$\int_0^T \Psi \frac{1}{2m} \rho C_d \frac{\partial A}{\partial b_n} |V| \frac{dx}{dt} dt = \int_0^T \Psi \frac{1}{2m} \rho C_d \left(\frac{\pi}{6}\right)^{1/3} \left(\frac{0.5}{\rho_1} + \frac{E}{\rho_2}\right)^{-1/3} \frac{1}{\rho_2} \delta_3^n |V| \frac{dx}{dt} dt \quad (IX)$$

Ο 4^{ος} όρος, λόγω του παραγωγισμού του μέτρου της ταχύτητας, πρέπει να αναλυθεί περαιτέρω σε 2 όρους.

$$= \int_0^T \Psi \frac{1}{2m} \rho C_d A \frac{\partial |V|}{\partial b_n} \frac{dx}{dt} dt + \int_0^T \Psi \frac{1}{2m} \rho C_d A \frac{1}{|V|} \frac{dx}{dt} \frac{dy}{dt} \frac{\partial}{\partial b_n} \left(\frac{dy}{dt}\right) dt \quad (X)$$

Ο 1^{ος} όρος της εξίσωσης (X) αναπτύσσεται ως εξής:

$$\begin{aligned} \int_0^T \Psi \frac{1}{2m} \rho C_d A \frac{1}{|V|} \left(\frac{dx}{dt}\right)^2 \frac{\partial}{\partial b_n} \left(\frac{dx}{dt}\right) dt &= \left[\Psi \frac{1}{2m} \rho C_d A \frac{1}{|V|} \left(\frac{dx}{dt}\right)^2 \frac{\partial x}{\partial b_n} \right]_0^T \\ &\quad - \int_0^T \frac{d\Psi}{dt} \frac{1}{2m} \rho C_d A \frac{1}{|V|} \left(\frac{dx}{dt}\right)^2 \frac{\partial x}{\partial b_n} dt \\ &\quad + \int_0^T \Psi \frac{1}{2m} \rho C_d A \frac{1}{|V|^3} \left(\frac{dx}{dt} \left(\frac{d^2x}{dt^2}\right) + \frac{dy}{dt} \left(\frac{d^2y}{dt^2}\right) \right) \left(\frac{dx}{dt}\right)^2 \frac{\partial x}{\partial b_n} dt \\ &\quad - \int_0^T \Psi \frac{1}{2m} \rho C_d A \frac{2}{|V|} \frac{dx}{dt} \left(\frac{d^2x}{dt^2}\right) \frac{\partial x}{\partial b_n} dt \end{aligned} \quad (XI)$$

Ο 2^{ος} όρος αναπτύσσεται με παρόμοιο τρόπο.

$$\begin{aligned} \int_0^T \Psi \frac{1}{2m} \rho C_d A \frac{1}{|V|} \frac{dx}{dt} \frac{dy}{dt} \frac{\partial}{\partial b_n} \left(\frac{dy}{dt}\right) dt &= \left[\Psi \frac{1}{2m} \rho C_d A \frac{1}{|V|} \frac{dx}{dt} \frac{dy}{dt} \frac{\partial y}{\partial b_n} \right]_0^T - \\ \int_0^T \frac{d\Psi}{dt} \frac{1}{2m} \rho C_d A \frac{1}{|V|} \frac{dx}{dt} \frac{dy}{dt} \frac{\partial y}{\partial b_n} dt &+ \int_0^T \Psi \frac{1}{2m} \rho C_d A \frac{1}{|V|^3} \left(\frac{dx}{dt} \left(\frac{d^2x}{dt^2}\right) + \frac{dy}{dt} \left(\frac{d^2y}{dt^2}\right) \right) \frac{dx}{dt} \frac{dy}{dt} \frac{\partial y}{\partial b_n} dt - \\ \int_0^T \Psi \frac{1}{2m} \rho C_d A \frac{1}{|V|} \frac{dy}{dt} \left(\frac{d^2x}{dt^2}\right) \frac{\partial y}{\partial b_n} dt &- \int_0^T \Psi \frac{1}{2m} \rho C_d A \frac{1}{|V|} \frac{dx}{dt} \left(\frac{d^2y}{dt^2}\right) \frac{\partial y}{\partial b_n} dt \end{aligned} \quad (XII)$$

Ο τελευταίος όρος της εξίσωσης (VI) αναλύεται ως εξής:

$$\begin{aligned} \int_0^T \Psi \frac{1}{2m} \rho C_d A |V| \frac{\partial}{\partial b_n} \left(\frac{dx}{dt}\right) dt &= \left[\Psi \frac{1}{2m} \rho C_d A |V| \frac{\partial x}{\partial b_n} \right]_0^T - \int_0^T \frac{d\Psi}{dt} \frac{1}{2m} \rho C_d A |V| \frac{\partial x}{\partial b_n} dt - \\ \int_0^T \Psi \frac{1}{2m} \rho C_d A \frac{1}{|V|} \left(\frac{dx}{dt} \left(\frac{d^2x}{dt^2}\right) + \frac{dy}{dt} \left(\frac{d^2y}{dt^2}\right) \right) \frac{\partial x}{\partial b_n} dt &\end{aligned} \quad (XIII)$$

Η ανάλυση του 3^{ου} όρου της εξίσωσης (IV) είναι σχεδόν η ίδια με αυτή του 2^{ου} όρου. Η ύπαρξη της επιτάχυνσης της βαρύτητας δεν επηρεάζει καθόλου την ανάλυση εφόσον δεν μεταβάλλεται ούτε με το χρόνο ούτε με τις μεταβλητές σχεδιασμού.

$$\frac{\delta}{\delta b_n} \int_0^T \Phi \left(\frac{d^2 y}{dt^2} + g + \frac{1}{2m} \rho C_d A |V| \frac{dy}{dt} \right) dt = \int_0^T \Phi \frac{\partial}{\partial b_n} \left(\frac{d^2 y}{dt^2} + g + \frac{1}{2m} \rho C_d A |V| \frac{dy}{dt} \right) dt + \left. \Phi \left(\frac{d^2 y}{dt^2} + g + \frac{1}{2m} \rho C_d A |V| \frac{dy}{dt} \right) \right|_T \frac{\delta T}{\delta b_n} - \left. \Phi \left(\frac{d^2 y}{dt^2} + g + \frac{1}{2m} \rho C_d A |V| \frac{dy}{dt} \right) \right|_0 \frac{\delta(0)}{\delta b_n} \quad (XIV)$$

Από την εξίσωση (XIV) μένει ο 1^{ος} όρος ο οποίος αναλύεται ως εξής:

$$\begin{aligned} \int_0^T \Phi \frac{\partial}{\partial b_n} \left(\frac{d^2 y}{dt^2} + g + \frac{1}{2m} \rho C_d A |V| \frac{dy}{dt} \right) dt &= \int_0^T \Phi \frac{\partial}{\partial b_n} \left(\frac{d^2 y}{dt^2} \right) dt + \\ &\int_0^T \Phi \frac{\partial}{\partial b_n} \left(\frac{1}{2m} \right) \rho C_d A |V| \frac{dy}{dt} dt + \int_0^T \Phi \frac{1}{2m} \rho C_d \frac{\partial A}{\partial b_n} |V| \frac{dy}{dt} dt + \\ &\int_0^T \Phi \frac{1}{2m} \rho C_d A \frac{\partial |V|}{\partial b_n} \frac{dy}{dt} dt + \int_0^T \Phi \frac{1}{2m} \rho C_d A |V| \frac{\partial}{\partial b_n} \left(\frac{dy}{dt} \right) dt \end{aligned} \quad (XV)$$

Για τον 1^ο όρο ισχύει ότι:

$$\int_0^T \Phi \frac{\partial}{\partial b_n} \left(\frac{d^2 y}{dt^2} \right) dt = \Phi(T) \frac{\partial}{\partial b_n} \left(\frac{dy}{dt} \right) \Big|_T - \Phi(0) \delta_2^n - \frac{d\Phi}{dt} \Big|_T \frac{\partial y(T)}{\partial b_n} + \int_0^T \frac{\partial y}{\partial b_n} \left(\frac{d^2 \Phi}{dt^2} \right) dt \quad (XVI)$$

Για τον 2^ο όρο έχουμε ότι:

$$\int_0^T \Phi \frac{\partial}{\partial b_n} \left(\frac{1}{2m} \right) \rho C_d A |V| \frac{dy}{dt} dt = - \int_0^T \Phi \frac{1}{2m^2} \delta_3^n \rho C_d A |V| \frac{dy}{dt} dt \quad (XVII)$$

Ο 3^{ος} όρος αναλύεται ως εξής:

$$\int_0^T \Phi \frac{1}{2m} \rho C_d \frac{\partial A}{\partial b_n} |V| \frac{dy}{dt} dt = \int_0^T \Phi \frac{1}{2m} \rho C_d \left(\frac{\pi}{6} \right)^{1/3} \left(\frac{0.5}{\rho_1} + \frac{E}{\rho_2} \right)^{-1/3} \frac{1}{\rho_2} \delta_3^n |V| \frac{dy}{dt} dt \quad (XVIII)$$

Ο 4^{ος} όρος αναλύεται περαιτέρω σε 2 όρους.

$$\begin{aligned} &\int_0^T \Phi \frac{1}{2m} \rho C_d A \frac{\partial |V|}{\partial b_n} \frac{dy}{dt} dt \\ = &\int_0^T \Phi \frac{1}{2m} \rho C_d A \frac{1}{|V|} \left(\frac{dy}{dt} \right)^2 \frac{\partial}{\partial b_n} \left(\frac{dy}{dt} \right) dt + \int_0^T \Phi \frac{1}{2m} \rho C_d A \frac{1}{|V|} \frac{dx}{dt} \frac{dy}{dt} \frac{\partial}{\partial b_n} \left(\frac{dx}{dt} \right) dt \end{aligned} \quad (XIX)$$

Οι επιμέρους όροι της εξίσωσης (XIX) αναπτύσσονται σύμφωνα με τις εξισώσεις (XX) και (XXI) αντίστοιχα.

$$\begin{aligned} \int_0^T \Phi \frac{1}{2m} \rho C_d A \frac{1}{|V|} \left(\frac{dy}{dt} \right)^2 \frac{\partial}{\partial b_n} \left(\frac{dy}{dt} \right) dt &= \left[\Phi \frac{1}{2m} \rho C_d A \frac{1}{|V|} \left(\frac{dy}{dt} \right)^2 \frac{\partial y}{\partial b_n} \right]_0^T \\ - \int_0^T \frac{d\Phi}{dt} \frac{1}{2m} \rho C_d A \frac{1}{|V|} \left(\frac{dy}{dt} \right)^2 \frac{\partial y}{\partial b_n} dt &+ \int_0^T \Phi \frac{1}{2m} \rho C_d A \frac{1}{|V|^3} \left(\frac{dx}{dt} \left(\frac{d^2 x}{dt^2} \right) + \frac{dy}{dt} \left(\frac{d^2 y}{dt^2} \right) \right) \left(\frac{dy}{dt} \right)^2 \frac{\partial y}{\partial b_n} dt \\ - \int_0^T \Phi \frac{1}{2m} \rho C_d A \frac{2}{|V|} \frac{dy}{dt} \left(\frac{d^2 y}{dt^2} \right) \frac{\partial y}{\partial b_n} dt & \quad (XX) \end{aligned}$$

$$\begin{aligned} \int_0^T \Phi \frac{1}{2m} \rho C_d A \frac{1}{|V|} \frac{dx}{dt} \frac{dy}{dt} \frac{\partial}{\partial b_n} \left(\frac{dx}{dt} \right) dt &= \left[\Phi \frac{1}{2m} \rho C_d A \frac{1}{|V|} \frac{dx}{dt} \frac{dy}{dt} \frac{\partial x}{\partial b_n} \right]_0^T - \\ \int_0^T \frac{d\Phi}{dt} \frac{1}{2m} \rho C_d A \frac{1}{|V|} \frac{dx}{dt} \frac{dy}{dt} \frac{\partial x}{\partial b_n} dt &+ \int_0^T \Phi \frac{1}{2m} \rho C_d A \frac{1}{|V|^3} \left(\frac{dx}{dt} \left(\frac{d^2 x}{dt^2} \right) + \frac{dy}{dt} \left(\frac{d^2 y}{dt^2} \right) \right) \frac{dx}{dt} \frac{dy}{dt} \frac{\partial x}{\partial b_n} dt - \\ \int_0^T \Phi \frac{1}{2m} \rho C_d A \frac{1}{|V|} \frac{dy}{dt} \left(\frac{d^2 x}{dt^2} \right) \frac{\partial x}{\partial b_n} dt &- \int_0^T \Phi \frac{1}{2m} \rho C_d A \frac{1}{|V|} \frac{dx}{dt} \left(\frac{d^2 y}{dt^2} \right) \frac{\partial x}{\partial b_n} dt \quad (XXI) \end{aligned}$$

Ο τελευταίος όρος της εξίσωσης (XV) αναλύεται παρακάτω.

$$\begin{aligned} \int_0^T \Phi \frac{1}{2m} \rho C_d A |V| \frac{\partial}{\partial b_n} \left(\frac{dy}{dt} \right) dt &= \left[\Phi \frac{1}{2m} \rho C_d A |V| \frac{\partial y}{\partial b_n} \right]_0^T - \int_0^T \frac{d\Phi}{dt} \frac{1}{2m} \rho C_d A |V| \frac{\partial y}{\partial b_n} dt - \\ \int_0^T \Phi \frac{1}{2m} \rho C_d A \frac{1}{|V|} \left(\frac{dx}{dt} \left(\frac{d^2 x}{dt^2} \right) + \frac{dy}{dt} \left(\frac{d^2 y}{dt^2} \right) \right) \frac{\partial y}{\partial b_n} dt & \quad (XXII) \end{aligned}$$

Ο 4^{ος} όρος της εξίσωσης (IV) παραγωγίζεται ως εξής:

$$\frac{\delta}{\delta b_n} (M(x(T) - x_{tar})^2) = 2M(x(T) - x_{tar}) \left(\frac{\partial x(T)}{\partial b_n} + \frac{\partial x(T)}{\partial T} \frac{\delta T}{\delta b_n} \right) \quad (XXIII)$$

Ομοίως, για τον 5^ο όρο της εξίσωσης (IV) ισχύει ότι:

$$\frac{\delta}{\delta b_n} (Qy(T)) = Q \left(\frac{\partial y(T)}{\partial b_n} + \frac{\partial y(T)}{\partial T} \frac{\delta T}{\delta b_n} \right) \quad (XXIV)$$

Τέλος, ο 1^{ος} όρος της (IV) δίνει τον όρο: $\delta T / \delta b_n$.

Από τις παραπάνω εξισώσεις, εξάγουμε τις FAE για τους αγνώστους $\partial x / \partial b_n$ και $\partial y / \partial b_n$ καθώς και τις οριακές συνθήκες για την επίλυσή τους. Η FAE για το $\partial x / \partial b_n$ παρουσιάζεται παρακάτω.

$$\begin{aligned} \frac{d^2 \Psi}{dt^2} &= \frac{1}{2m} \rho C_d A \left(\frac{1}{|V|} \left(\frac{dx}{dt} \right)^2 + |V| \right) \frac{d\Psi}{dt} \\ &- \frac{1}{2m} \rho C_d A \frac{1}{|V|} \left(-\frac{1}{|V|^2} \left(\frac{dx}{dt} \left(\frac{d^2 x}{dt^2} \right) + \frac{dy}{dt} \left(\frac{d^2 y}{dt^2} \right) \right) \left(\frac{dx}{dt} \right)^2 + 3 \frac{dx}{dt} \left(\frac{d^2 x}{dt^2} \right) + \frac{dy}{dt} \left(\frac{d^2 y}{dt^2} \right) \right) \Psi \\ &- \frac{1}{2m} \rho C_d A \frac{1}{|V|} \frac{dx}{dt} \frac{dy}{dt} \frac{\partial x}{\partial b_n} \frac{d\Phi}{dt} \\ &- \frac{1}{2m} \rho C_d A \frac{1}{|V|} \left(-\frac{1}{|V|^2} \left(\frac{dx}{dt} \left(\frac{d^2 x}{dt^2} \right) + \frac{dy}{dt} \left(\frac{d^2 y}{dt^2} \right) \right) \frac{dx}{dt} \frac{dy}{dt} + \frac{dy}{dt} \left(\frac{d^2 x}{dt^2} \right) + \frac{dx}{dt} \left(\frac{d^2 y}{dt^2} \right) \right) \Phi \quad (XXV) \end{aligned}$$

Η FAE για το $\partial y / \partial b_n$ προκύπτει από την εξίσωση (XXV) αλλάζοντας παντού το Φ με το Ψ , όπως φαίνεται στην εξίσωση (XXVI).

$$\begin{aligned} \frac{d^2 \Phi}{dt^2} &= \frac{1}{2m} \rho C_d A \left(\frac{1}{|V|} \left(\frac{dy}{dt} \right)^2 + |V| \right) \frac{d\Phi}{dt} \\ &- \frac{1}{2m} \rho C_d A \frac{1}{|V|} \left(-\frac{1}{|V|^2} \left(\frac{dx}{dt} \left(\frac{d^2 x}{dt^2} \right) + \frac{dy}{dt} \left(\frac{d^2 y}{dt^2} \right) \right) \left(\frac{dy}{dt} \right)^2 + \frac{dx}{dt} \left(\frac{d^2 x}{dt^2} \right) + 3 \frac{dy}{dt} \left(\frac{d^2 y}{dt^2} \right) \right) \Phi \\ &- \frac{1}{2m} \rho C_d A \frac{1}{|V|} \frac{dx}{dt} \frac{dy}{dt} \frac{\partial x}{\partial b_n} \frac{d\Psi}{dt} \\ &- \frac{1}{2m} \rho C_d A \frac{1}{|V|} \left(-\frac{1}{|V|^2} \left(\frac{dx}{dt} \left(\frac{d^2 x}{dt^2} \right) + \frac{dy}{dt} \left(\frac{d^2 y}{dt^2} \right) \right) \frac{dx}{dt} \frac{dy}{dt} + \frac{dy}{dt} \left(\frac{d^2 x}{dt^2} \right) + \frac{dx}{dt} \left(\frac{d^2 y}{dt^2} \right) \right) \Psi \quad (XXVI) \end{aligned}$$

Οι οριακές συνθήκες για την επίλυση των FAE προκύπτουν από τους όρους τους οποίους δεν θέλουμε να υπολογίσουμε. Αυτοί είναι το $\delta T / \delta b_n$, το $\partial x(T) / \partial b_n$, $\frac{\partial}{\partial b_n} \left(\frac{dy}{dt} \right) \Big|_T$, $\frac{\partial}{\partial b_n} \left(\frac{dx}{dt} \right) \Big|_T$ και το $\partial y(T) / \partial b_n$. Επομένως, οι ABC είναι οι ακόλουθες.

$$\left\{ \begin{array}{l} \Phi(T) = 0 \\ \Psi(T) = 0 \\ \frac{d\Psi}{dt} \Big|_T = 2M(X(T) - x_{tar}) \\ \frac{d\Phi}{dt} \Big|_T = -\frac{1 - 2M(X(T) - x_{tar}) \partial x(T) / \partial T}{\partial y(T) / \partial T} \end{array} \right. \quad (XXVII)$$

Από τις ABC, είναι προφανές ότι η επίλυση των FAE θα γίνει ανάποδα. Επιλύοντας τις FAE βρίσκουμε τις παραγώγους ευαισθησίας, οι οποίες παρουσιάζονται παρακάτω.

$$\left\{ \begin{array}{l} \frac{\delta F}{\delta b_1} = -\Psi(0) \\ \frac{\delta F}{\delta b_2} = -\Phi(0) \\ \frac{\delta F}{\delta b_3} = -\int_0^T \left(\Psi \frac{dx}{dt} + \Phi \frac{dy}{dt} \right) \frac{1}{2m^2} \rho C_d A |V| dt + \int_0^T \left(\Psi \frac{dx}{dt} + \Phi \frac{dy}{dt} \right) \frac{1}{2m} \rho C_d \left(\frac{\pi}{6} \right)^{1/3} \left(\frac{0.5}{\rho_1} + \frac{E}{\rho_2} \right)^{-1/3} \frac{1}{\rho_2} |V| dt \end{array} \right. \quad (XXVIII)$$

Με τις παραγώγους από την σχέση (XXVIII) εφαρμόζεται η μέθοδος Steepest Descent (SD).

2.3 Διαχείριση περιορισμού κάθετου τοίχου

Εκτός από το περιορισμό η μπάλα να φτάσει στο σημείο $(x,y)=(20,0)$, πρέπει να περάσει και μέσα από ένα τοίχο που βρίσκεται στο $x_w=10m$ με άνοιγμα από $y_{min}=8m$ έως και $y_{max}=11m$. Αυτός ο περιορισμός θα αντιμετωπιστεί με την μέθοδο ALM όπως έγινε και με το περιορισμό για τον άξονα x με την διαφορά ότι δεν μπορεί να συμπεριληφθεί στις ABC με ορθό τρόπο. Αυτό συμβαίνει διότι ο περιορισμός λειτουργεί για κάποιο χρόνο T_w όπου η μπάλα περνάει από τον τοίχο και όχι για T όπου πέφτει στο έδαφος. Για την επίλυση αυτού του προβλήματος «τσοντάρουμε» το περιορισμό στην οριακή συνθήκη του $d\Phi/dt$. Η εξίσωση (XXIX) αντιστοιχεί στον περιορισμό.

$$C = \begin{cases} 2M(y_{wall} - R - y_{min}), y_{wall} - R < y_{min} \\ 2M(y_{wall} + R - y_{max}), y_{wall} + R > y_{max} \\ 0, y_{min} < y_{wall} + R < y_{max} \end{cases} \quad (XXIX)$$

Λαμβάνοντας υπόψιν το περιορισμό, οι ABC αλλάζουν ως εξής:

$$\begin{aligned} \Phi(T) &= 0 \\ \Psi(T) &= 0 \\ \frac{d\Psi}{dt} \Big|_T &= 2M(X(T) - x_{tar}) \\ \frac{d\Phi}{dt} \Big|_T &= -\frac{1 - 2M(X(T) - x_{tar}) \partial x(T)/\partial T}{\partial y(T)/\partial T} + C \end{aligned} \quad (XXVII')$$

Παρόλο που αυτό το τέχνασμα δεν έχει μαθηματικό υπόβαθρο, είναι αποδεκτό καθώς οδηγεί σε σύγκλιση της μεθόδου.

3 Μέθοδος πεπερασμένων διαφορών (FD)

Η μέθοδος των πεπερασμένων διαφορών είναι μια εναλλακτική μέθοδος υπολογισμού των παραγώγων ευαισθησίας η οποία βασίζεται στην επίλυση του primal προβλήματος πολλές φορές. Πιο συγκεκριμένα, οι παράγωγοι υπολογίζονται από την εξίσωση (XXX).

$$\begin{aligned}\frac{\delta F}{\delta b_1} &= \frac{F(b_1 + \varepsilon, b_2, b_3) - F(b_1 - \varepsilon, b_2, b_3)}{2\varepsilon} \\ \frac{\delta F}{\delta b_2} &= \frac{F(b_1, b_2 + \varepsilon, b_3) - F(b_1, b_2 - \varepsilon, b_3)}{2\varepsilon} \\ \frac{\delta F}{\delta b_3} &= \frac{F(b_1, b_2, b_3 + \varepsilon) - F(b_1, b_2, b_3 - \varepsilon)}{2\varepsilon}\end{aligned}\tag{XXX}$$

Η συνάρτηση που εμφανίζεται στην παραπάνω εξίσωση είναι ίδια με την εξίσωση (IV) με τη διαφορά ότι δεν συμπεριλαμβάνει τα 2 ολοκληρώματα. Το ε είναι ένας πολύ μικρός αριθμός ο οποίος επηρεάζει σημαντικά τις παραγώγους.

4 Μέθοδος συζυγών μεταβλητών (CV)

Τελευταία μέθοδος που θα αναλυθεί είναι η μέθοδος των συζυγών μεταβλητών. Σύμφωνα με την μέθοδο αυτή, οι παράγωγοι ευαισθησίας υπολογίζονται από την σχέση (XXXI).

$$\begin{aligned}\frac{\delta F}{\delta b_1} &= \frac{\text{imag}F(b_1 + \varepsilon, b_2, b_3)}{\varepsilon} \\ \frac{\delta F}{\delta b_2} &= \frac{\text{imag}F(b_1, b_2 + \varepsilon, b_3)}{\varepsilon} \\ \frac{\delta F}{\delta b_3} &= \frac{\text{imag}F(b_1, b_2, b_3 + \varepsilon)}{\varepsilon}\end{aligned}\tag{XXXI}$$

Συγκρίνοντας τις εξισώσεις (XXIX) και (XXX), φαίνεται ότι οι 2 μέθοδοι είναι παρόμοιες. Οι διαφορές μεταξύ τους έγκειται στο ότι η CV παρουσιάζει πολύ μικρότερη εξάρτηση από το ε σε σχέση με την FD. Επιπλέον, για το ίδιο επίπεδο ακρίβειας, η CV χρειάζεται να λύσει το primal πρόβλημα τις μισές φορές σε σχέση με το FD.

Παρόλο όμως που η μέθοδος CV θεωρείται ότι είναι καλύτερη, στη παρούσα περίπτωση έχει ένα μεγάλο πρόβλημα. Ο χρόνος T που υπάρχει στην συνάρτηση F , ο οποίος παίζει σημαντικό ρόλο στην διαδικασία της βελτιστοποίησης, δεν μπορεί να έχει φανταστικό μέρος. Αυτό δημιουργεί ένα μεγάλο πρόβλημα καθώς τότε οι παράγωγοι ευαισθησίας δεν επηρεάζονται από το μέγεθος που θέλουμε να ελαχιστοποιήσουμε αλλά μόνο από τους περιορισμούς. Εφόσον ο περιορισμός εξαρτάται από το φανταστικό μέρος του $x(T)$, η μέθοδος προσπαθεί να το ελαχιστοποιήσει. Αυτό έχει ως αποτέλεσμα το πραγματικό μέρος του $x(T)$ συνεχώς να μειώνεται παραβιάζοντας το περιορισμό του άξονα x . Μια πιθανή λύση για αυτό είναι να γίνει παρεμβολή ώστε το $y(T)$ να είναι

ακριβώς μηδέν. Τότε ο χρόνος T θα είχε φανταστικό μέρος. Δυστυχώς, αυτή η μέθοδος δεν είχε αποτέλεσμα.

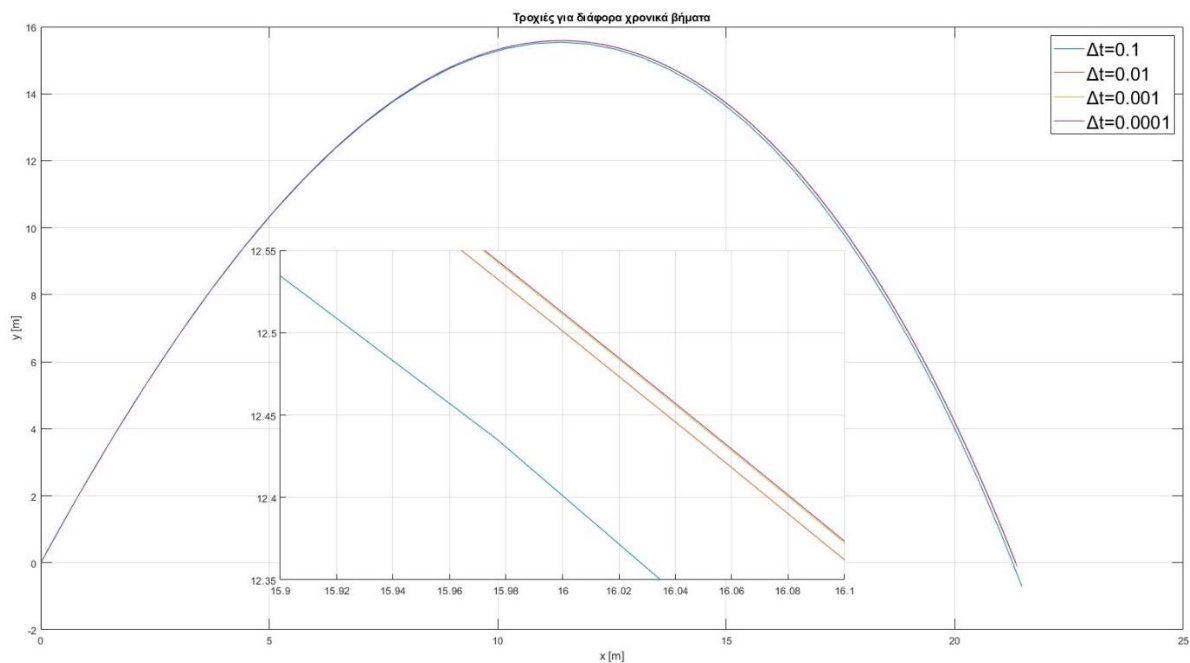
Μια άλλη πιθανή λύση είναι στις παραγώγους της εξίσωσης (XXXI) να προστεθεί επιπλέον ο όρος $Re(x(T)) - x_{tar}$. Με αυτό τον τρόπο ποινικοποιείται η λύση όταν φεύγει μακριά από το x_{tar} . Στην πράξη, και αυτή η μέθοδος δεν δίνει το επιθυμητό αποτέλεσμα καθώς όσο μεγαλώνει ο συντελεστής M της ALM, το $X(T)$ αρχίζει να ταλαντώνεται γύρω από το x_{tar} .

Όταν όμως συμπεριληφθεί ο περιορισμός του ενδιαμέσου τοίχου, όπως αναφέρθηκε παραπάνω, η μέθοδος συγκλίνει χωρίς όμως να ελαχιστοποιεί πραγματικά το T .

5 Αριθμητικά αποτελέσματα

5.1 Επιλογή βήματος Runge-Kutta

Για την επιλογή χρονικού βήματος της Runge-Kutta έγιναν δοκιμές για μερικά χρονικά βήματα. Όπως φαίνεται στο διάγραμμα 1 και ειδικά στο μεγεθυμένο σχήμα, η διαφορά μεταξύ των λύσεων για $\Delta t=0.001s$ και $\Delta t=0.0001s$ είναι αμελητέα. Για αυτό επιλέχτηκε χρονικό βήμα $\Delta t=0.001s$.



Διάγραμμα 1: Τροχιές για διάφορα χρονικά βήματα.

5.2 Σύγκριση παραγώγων ευαισθησίας για διάφορες τιμές του ε

Σε αυτό το κεφάλαιο θα γίνει η σύγκριση μεταξύ των παραγώγων που προκύπτουν από τις διάφορες μεθόδους. Η σύγκριση μεταξύ των μεθόδων γίνεται για την 1^η παράγωγο που υπολογίζεται με την ίδια αρχικοποίηση ($u_0=8$ m/s, $v_0=20$ m/s και $E=0.1$ kg). Στο πίνακα 1 παρουσιάζονται οι παράγωγοι που προκύπτουν από την μέθοδο FD.

ε	$\partial F / \partial b_1$	$\partial F / \partial b_2$	$\partial F / \partial b_3$
0.1	2.3507627	0.5676414	-22.9917473
0.001	2.2562951	1.9208405	-10.7727735
10^{-10}	2.2559096	1.9209596	-15.5554846
10^{-15}	1.9127061	0	-10.4683622

Πίνακας 1: Τιμές παραγώγων με μέθοδο FD.

Οι παράγωγοι της μεθόδου CV του πίνακα 2 υπολογίστηκαν λαμβάνοντας υπόψιν και το περιορισμό του κάθετου τοίχου καθώς μόνο τότε συγκλίνει η μέθοδος.

ε	$\partial F / \partial b_1$	$\partial F / \partial b_2$	$\partial F / \partial b_3$
0.1	1.3928124	13.5904132	-45.4782694
0.001	1.3896380	13.5901688	-57.3594313
10^{-10}	1.3896378	13.5901688	-57.3607047
10^{-15}	1.3896378	13.5901688	-57.3607047

Πίνακας 2: Τιμές παραγώγων με μέθοδο CV.

Συγκρίνοντας τις τιμές των 2 πινάκων είναι εμφανές ότι οι παράγωγοι που προκύπτουν από την CV όχι μόνο σταθεροποιούνται πιο γρήγορα από τις παραγώγους της FD αλλά επίσης δεν επηρεάζονται καθόλου όταν το ε παίρνει πολύ μικρές τιμές. Τέλος, στο πίνακα 3 παρατίθενται οι παράγωγοι που προκύπτουν από την adjoint.

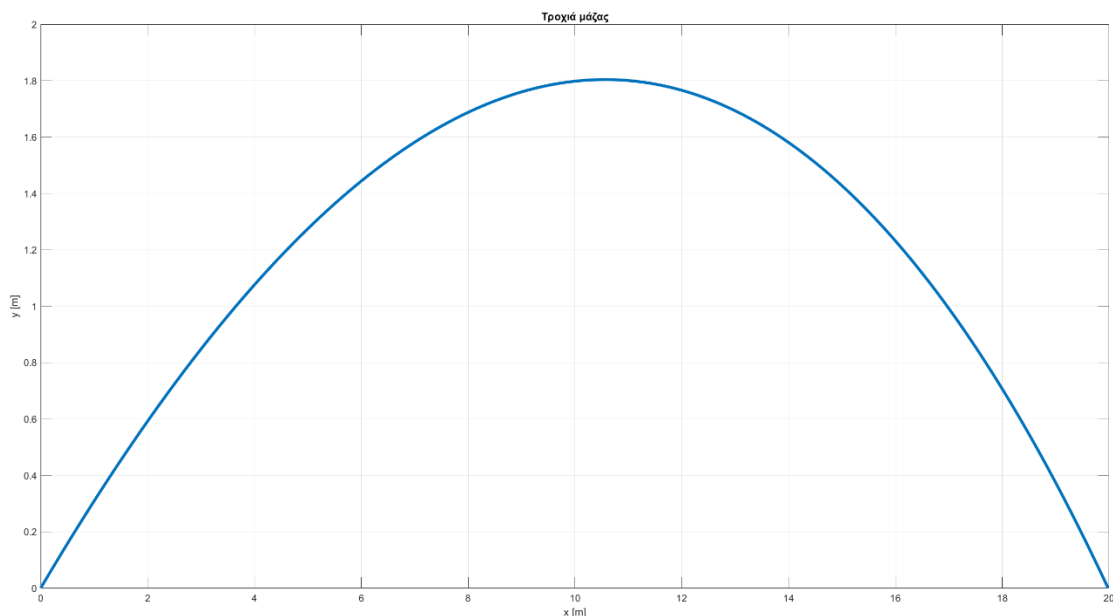
$\partial F / \partial b_1$	$\partial F / \partial b_2$	$\partial F / \partial b_3$
0.1045183	0.8746427	-4.0800848

Πίνακας 3: Παράγωγοι από τη συνεχή συζυγή μέθοδο.

Από τους παραπάνω πίνακες παρατηρούμε ότι παρόλο που οι παράγωγοι είναι αρκετά διαφορετικές μεταξύ τους, σε κάθε περίπτωση έχουν το ίδιο πρόσημο. Δηλαδή, με όλες τις μεθόδους, το $\partial F / \partial b_3$ είναι αρνητικό. Αυτό σημαίνει ότι πιθανότατα δεν έχει γίνει λάθος στην υλοποίηση των μεθόδων.

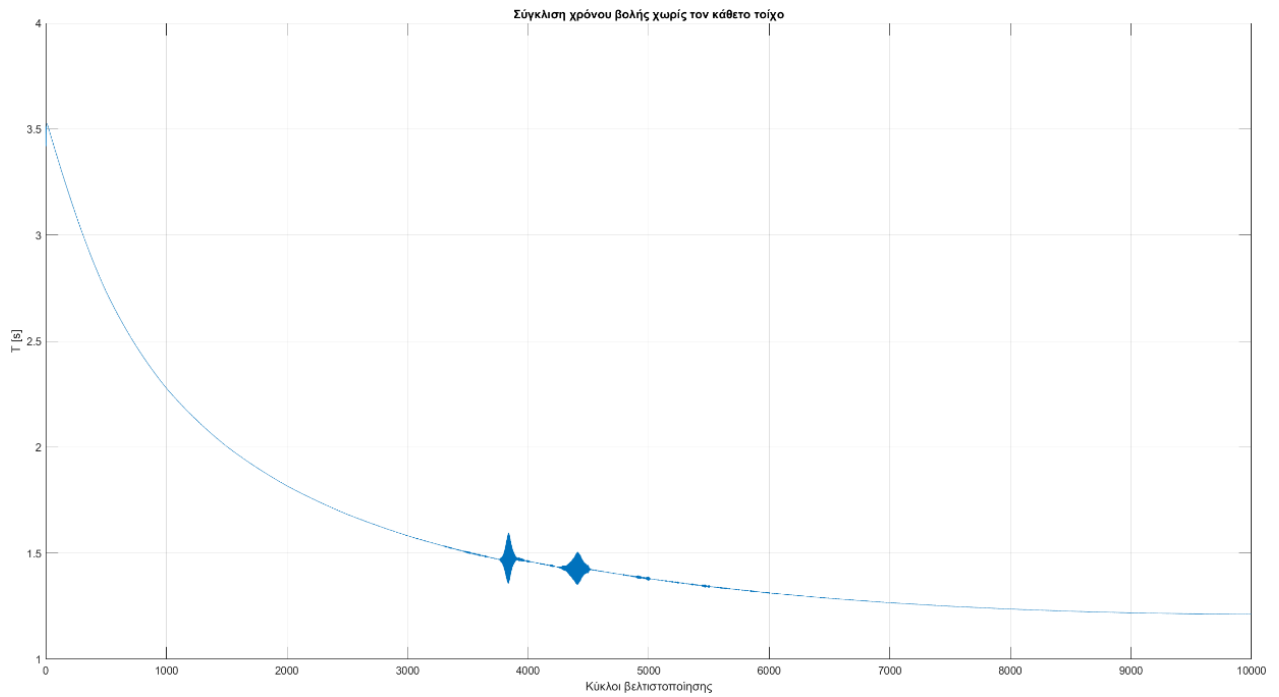
5.3 Βελτιστοποίηση με adjoint χωρίς το περιορισμό του κάθετου τοίχου

Η απουσία του κάθετου τοίχου επιτρέπει την ύπαρξη θεωρητικά άπειρων βέλτιστων λύσεων. Ανάλογα με την αρχικοποίηση των μεταβλητών σχεδιασμού, καταλήγουμε σε σχεδόν ίδιο χρόνο T με διάφορους συνδυασμούς των μεταβλητών σχεδιασμού. Στο διάγραμμα 2 φαίνεται η τροχιά που προκύπτει για αρχικές συνθήκες $b = [8 \ 20 \ 0.1]$. Οι εσωτερικές επαναλήψεις του adjoint είναι 500 ενώ οι εξωτερικές για την ALM είναι 20. Ο συντελεστής M της ALM έχει αρχική τιμή 1.3 και αυξάνεται κατά 3% με κάθε επανάληψη. Το βήμα για την SD επιλέχτηκε έτσι ώστε στην αρχή να είναι μεγάλο και κάθε φορά που αυξάνεται το M να μειώνεται. Πιο συγκεκριμένα, $\eta = 0.25 - 0.012p$ όπου p ο αριθμός των επαναλήψεων για την λούπα της ALM. Αυτό το τρέξιμο δίνει τις τελικές μεταβλητές $b = [19.776562 \ 6.359729 \ 2.398608]$ με $T=1.2125$.



Διάγραμμα 2: Τροχιά μάζας χωρίς τον κάθετο τοίχο με μέθοδο adjoint.

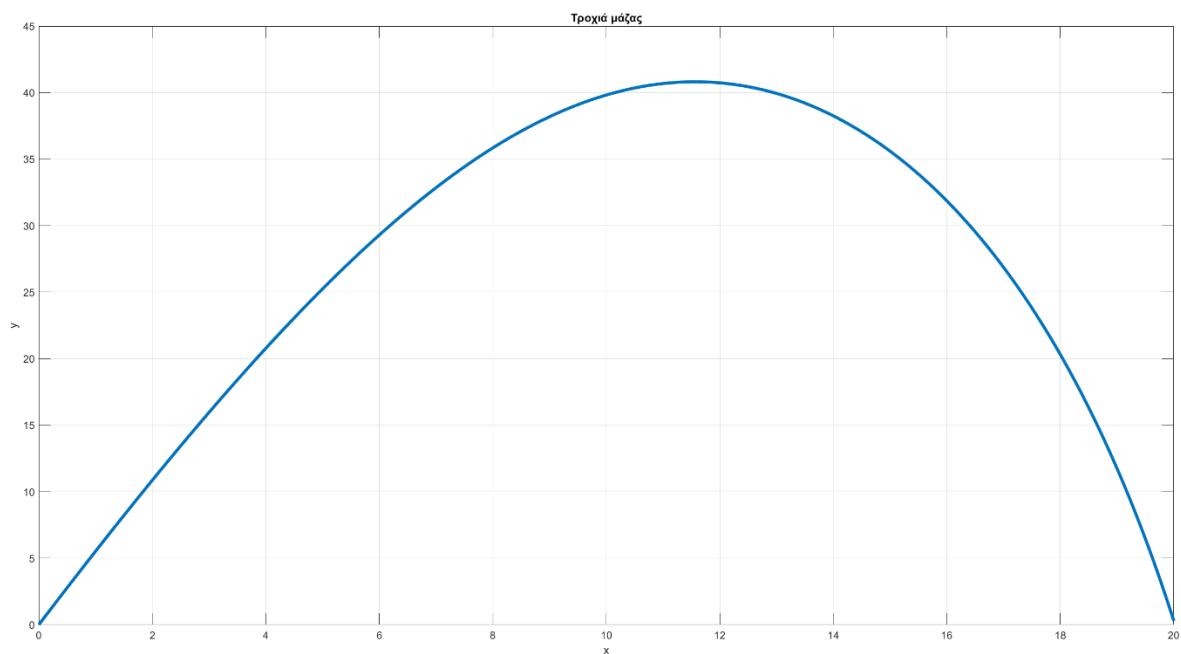
Για αυτή την περίπτωση ο χρόνος T καταλήγει σε μια σταθερή τιμή κοντά στους 9000 κύκλους βελτιστοποίησης όπως φαίνεται στο διάγραμμα 3.



Διάγραμμα 3: Σύγκλιση χρόνου T συναρτήσει κύκλων βελτιστοποίησης με μέθοδο *adjoint*.

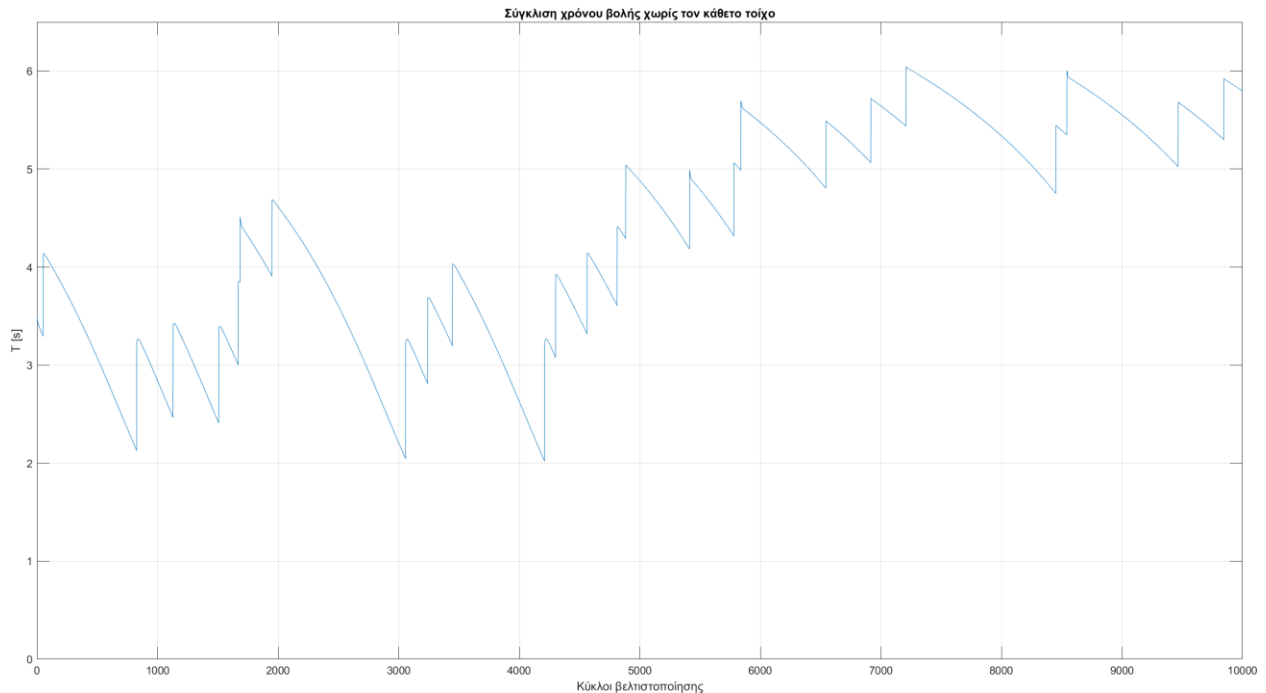
5.4 Βελτιστοποίηση με FD χωρίς το περιορισμό του κάθετου τοίχου

Για την βελτιστοποίηση με FD χρησιμοποιούμε τα ίδια δεδομένα με αυτά που χρησιμοποιήθηκαν προηγουμένως για την *adjoint* με την διαφορά ότι το βήμα μειώθηκε από 0.1 σε 0.01. Από το διάγραμμα 4 παρατηρούμε ότι παρόλο που έγιναν τόσες πολλές επαναλήψεις, το αποτέλεσμα είναι μακριά από το βέλτιστο.



Διάγραμμα 4: Τροχιά μάζας με μέθοδο FD.

Το διάγραμμα 5 μας δείχνει ότι παρόλο που γίνονται πολλές εσωτερικές επαναλήψεις, λόγω της χαμηλής ακρίβειας της μεθόδου και πιθανότατα του μικρού βήματος, δεν φτάνουμε ποτέ στο βέλτιστο. Το γεγονός ότι ο χρόνος T παίρνει χειρότερες τιμές σε τυχαία σημεία και όχι όταν αυξάνεται ο συντελεστής M επιβεβαιώνει ότι αυτή η μέθοδος δεν παράγει πολύ ακριβείς παραγώγους. Έτσι, δικαιολογείται και η τροχιά που προέκυψε στο διάγραμμα 4.

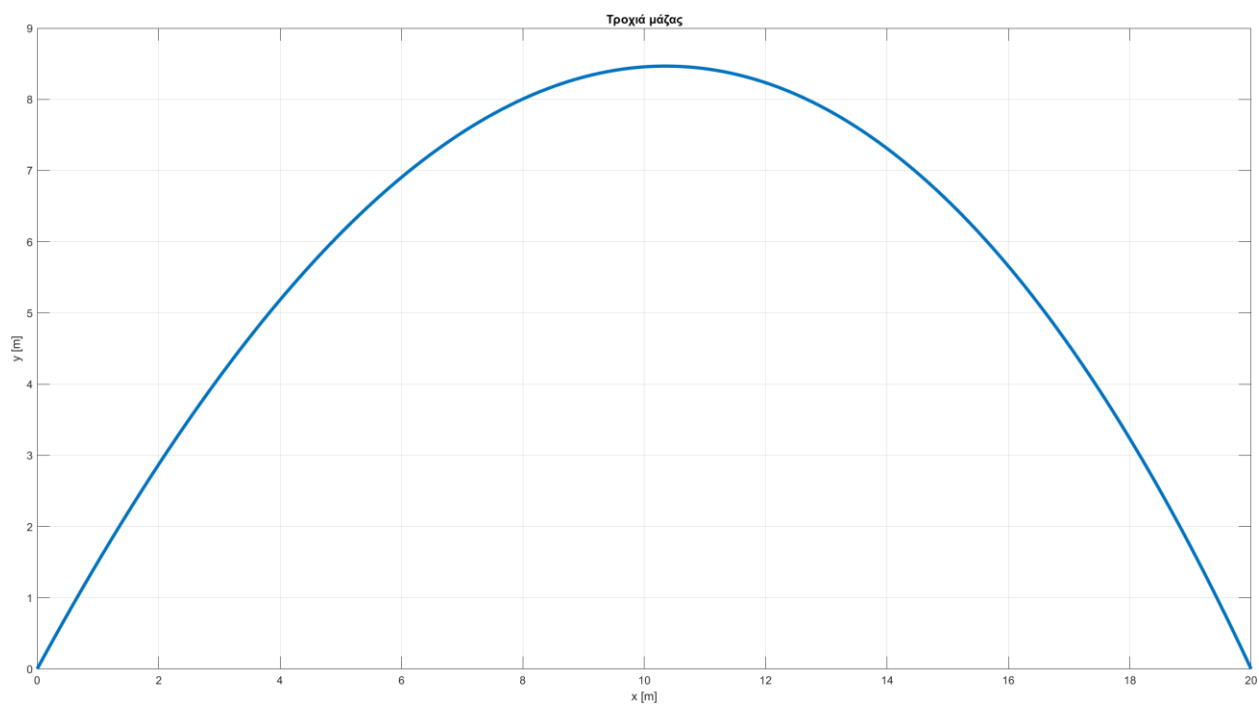


Διάγραμμα 5: Σύγκλιση χρόνου T συναρτήσει κύκλων βελτιστοποίησης με μέθοδο FD.

5.5 Βελτιστοποίηση με adjoint με το περιορισμό του κάθετου τοίχου

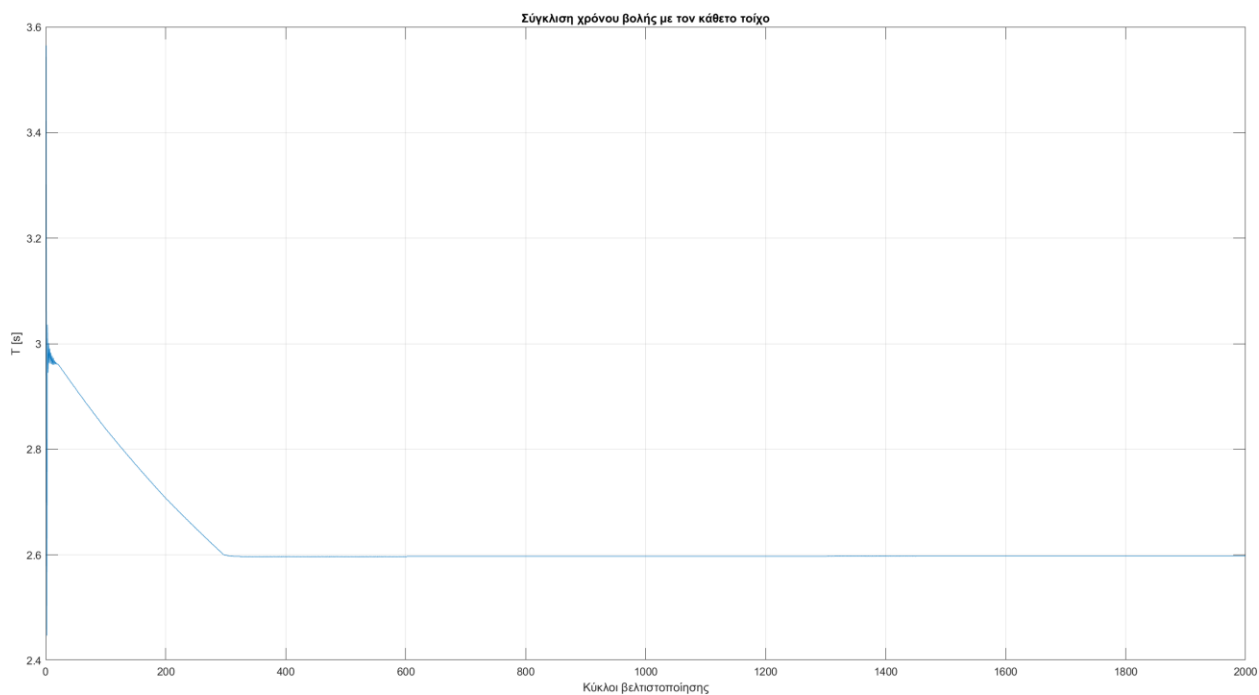
Για την περίπτωση όπου υπάρχει ο κάθετος τοίχος, έγιναν διάφορες δοκιμές αλλάζοντας την αρχικοποίηση του b και του M , τον αριθμό των εσωτερικών επαναλήψεων και της ποσοστιαίας αύξησης του M . Σε όλες αυτές τις περιπτώσεις, η μέθοδος σύγκλινε με πάρα πολύ λίγες επαναλήψεις. Πρακτικά σύγκλινε πριν καν τελειώσει η πρώτη επανάληψη της ALM και έπειτα δεν μετακινούνταν μέχρι το τέλος. Όταν όμως λέμε ότι σύγκλινε, δεν εννοούμε πραγματική σύγκλιση αλλά μια ταλάντωση ανάμεσα σε 2 λύσεις από τις οποίες η μια οριακά δεν ικανοποιεί τον περιορισμό και η άλλη οριακά τον ικανοποιεί. Λόγω της ταχύτητας της σύγκλισης, μειώσαμε τις εσωτερικές επαναλήψεις από 500 σε 100 χωρίς να αλλοιωθεί το αποτέλεσμα. Σε όλα τα τρεξίματα, η βέλτιστη λύση περνούσε οριακά από το κάτω τοίχωμα. Ένα πρόβλημα όμως της adjoint είναι η βελτιστοποίηση του E . Αλλάζοντας τις αρχικοποιήσεις των μεταβλητών σχεδιασμού, το E παίρνει διαφορετική τιμή κάθε φορά και τα u_0 και v_0 δρουν ως ρεγουλαριστές οι οποίοι ελαχιστοποιούν το χρόνο για το συγκεκριμένο E . Από τα τρεξίματα που έγιναν, η καλύτερη λύση επιτεύχθηκε χρησιμοποιώντας τις αρχικές συνθήκες που αναγράφονται στο κεφάλαιο 5.3 αλλάζοντας την αρχικοποίηση του E σε 5. Το διάγραμμα 6 παρουσιάζει την τροχιά της μάζας για αυτές τις αρχικοποιήσεις. Για $x=10m$, παρατηρούμε ότι πράγματι η μπάλα οριακά περνάει μέσα

από τον τοίχο. Αυτή η λύση δίνει τελικές μεταβλητές σχεδιασμού $b = [9.48929 \ 13.914294 \ 4.205266]$ με χρόνο $T=2.598s$.



Διάγραμμα 6: Τροχιά μάζας όταν υπάρχει ενδιάμεσος τοίχος με μέθοδο *adjoint*.

Στο διάγραμμα 7 παρατηρείται η ταχύτητα της σύγκλισης.



Διάγραμμα 7: Σύγκλιση χρόνου T συναρτήσει κύκλων βελτιστοποίησης με μέθοδο *adjoint*.

6 Παράρτημα κώδικα

6.1 Κώδικας για την FD

```
clc
clear
close all

format long g

xtar = 20;
r1 = 200;
r2 = 50;
rair = 1.225;
cd = 0.47 + 2/80;
g = 9.81;

t0 = 0;
tf = 10;
dt = 0.001;

x0 = 0;
y0 = 0;

psif = 0;
Ff = 0;

u0 = 8; %b1
v0 = 20; %b2
E = 0.1; %b3
M = 1.3;
epsilon = 10^(-5);
kill = false;
count = 1;

for p = 1:20 %ALM

    for k = 1:500
        R = (0.75/pi)^(1/3) * (0.5/r1 + E/r2)^(1/3);
        if imag(R) ~= 0
            kill = true;
            break;
        end
        A = pi * R^2;
        m = 0.5+E;

        F1 = @(u,v) -(2*m)^(-1)*rair*cd*A*sqrt(u^2 + v^2)*u;
        F2 = @(u,v) -g - (2*m)^(-1)*rair*cd*A*sqrt(u^2 + v^2)*v;

        % [x1,y1] = RK4(F1,F2,x0,y0,u0,v0,0.1,tf);
        % [x2,y2] = RK4(F1,F2,x0,y0,u0,v0,0.01,tf);
        % [x3,y3] = RK4(F1,F2,x0,y0,u0,v0,0.001,tf);
        % [x4,y4] = RK4(F1,F2,x0,y0,u0,v0,0.0001,tf);
        %
        % figure;
        % plot(x1,y1)
        % hold on
```

```

% plot(x2,y2)
% plot(x3,y3)
% plot(x4,y4)
% hold off
% grid on
% xlabel('x [m]')
% ylabel('y [m]')
% legend('Δt=0.1', 'Δt=0.01', 'Δt=0.001', 'Δt=0.0001')

[x,y,u,v,T,k1u,k1v,k2u,k2v,k3u,k3v,k4u,k4v] = RK4(F1,F2,x0,y0,u0,v0,dt,tf);
Tp(count)=T;
count=count+1;

[Df1,Df2] = FD(F1,F2,x0,y0,u0,v0,dt,tf,epsilon,M,xtar);

E = E + epsilon;
R = (0.75/pi)^(1/3) * (0.5/r1 + E/r2)^(1/3);
A = pi * R^2;
m = 0.5+E;
F1 = @(u,v) -(2*m)^(-1)*rair*cd*A*sqrt(u^2 + v^2)*u;
F2 = @(u,v) -g - (2*m)^(-1)*rair*cd*A*sqrt(u^2 + v^2)*v;

[x5,y5,~,~,T5] = RK4(F1,F2,x0,y0,u0,v0,dt,tf);

E = E - 2*epsilon;
R = (0.75/pi)^(1/3) * (0.5/r1 + E/r2)^(1/3);
A = pi * R^2;
m = 0.5+E;
F1 = @(u,v) -(2*m)^(-1)*rair*cd*A*sqrt(u^2 + v^2)*u;
F2 = @(u,v) -g - (2*m)^(-1)*rair*cd*A*sqrt(u^2 + v^2)*v;

[x6,y6,~,~,T6] = RK4(F1,F2,x0,y0,u0,v0,dt,tf);

Df3 = ((T5 - T6) + M*((x5(end)-xtar)^2 - (x6(end) -xtar)^2) + (y5(end) - y6(end)))/(2
* epsilon);

E = E + epsilon;

if E - 0.01 * Df3 < 0
    kill = true;
    break;
end
E = E - 0.01 * Df3;
u0 = u0 - 0.01 * Df1;
v0 = v0 - 0.01 * Df2;
end
if kill
    break;
end
if M < 30
    M = M*1.03;
end
end

function [Df1,Df2] = FD(F1,F2,x0,y0,u0,v0,step,tf,epsilon,M,xtar)
    Df1 = 0;
    Df2 = 0;
    [x1,y1,~,~,T1] = RK4(F1,F2,x0,y0,u0+epsilon,v0,step,tf);
    [x2,y2,~,~,T2] = RK4(F1,F2,x0,y0,u0-epsilon,v0,step,tf);

```

```

    Df1 = ((T1-T2) + M*((x1(end)-xtar)^2 - (x2(end) -xtar)^2) + (y1(end) -
y2(end)))/(2*epsilon);
    [x3,y3,~,~,T3] = RK4(F1,F2,x0,y0,u0,v0+epsilon,step,tf);
    [x4,y4,~,~,T4] = RK4(F1,F2,x0,y0,u0,v0-epsilon,step,tf);
    Df2 = ((T3-T4) + + M*((x3(end)-xtar)^2 - (x4(end) -xtar)^2) + (y3(end) -
y4(end)))/(2*epsilon);
end

```

```

function [x,y,u,v,T,k1u,k1v,k2u,k2v,k3u,k3v,k4u,k4v] = RK4(F1,F2,x0,y0,u0,v0,step,tf)

```

```

x = zeros(tf/step,1);
x(1) = x0;
y = zeros(tf/step,1);
y(1) = y0;
u = zeros(tf/step,1);
u(1) = u0; %b1
v = zeros(tf/step,1);
v(1) = v0; %b2
k1x = zeros(tf/step,1);
k2x = zeros(tf/step,1);
k3x = zeros(tf/step,1);
k4x = zeros(tf/step,1);

k1y = zeros(tf/step,1);
k2y = zeros(tf/step,1);
k3y = zeros(tf/step,1);
k4y = zeros(tf/step,1);

k1u = zeros(tf/step,1);
k2u = zeros(tf/step,1);
k3u = zeros(tf/step,1);
k4u = zeros(tf/step,1);

k1v = zeros(tf/step,1);
k2v = zeros(tf/step,1);
k3v = zeros(tf/step,1);
k4v = zeros(tf/step,1);

for i = 1:(tf/step - 1)

    k1x(i) = step * u(i);
    k1y(i) = step * v(i);
    k1u(i) = step * F1(u(i),v(i));
    k1v(i) = step * F2(u(i),v(i));

    k2x(i) = step * (u(i) + k1u(i)/2);
    k2y(i) = step * (v(i) + k1v(i)/2);
    k2u(i) = step * F1(u(i) + k1u(i)/2,v(i) + k1v(i)/2);
    k2v(i) = step * F2(u(i) + k1u(i)/2,v(i) + k1v(i)/2);

    k3x(i) = step * (u(i) + k2u(i)/2);
    k3y(i) = step * (v(i) + k2v(i)/2);
    k3u(i) = step * F1(u(i) + k2u(i)/2,v(i) + k2v(i)/2);
    k3v(i) = step * F2(u(i) + k2u(i)/2,v(i) + k2v(i)/2);

    k4x(i) = step * (u(i) + k3u(i));
    k4y(i) = step * (v(i) + k3v(i));
    k4u(i) = step * F1(u(i) + k3u(i)/2,v(i) + k3v(i)/2);
    k4v(i) = step * F2(u(i) + k3u(i)/2,v(i) + k3v(i)/2);

    x(i+1) = x(i) + (k1x(i) + 2*k2x(i) + 2*k3x(i) + k4x(i))/6;

```

```

y(i+1) = y(i) + (k1y(i) + 2*k2y(i) + 2*k3y(i) + k4y(i))/6;
u(i+1) = u(i) + (k1u(i) + 2*k2u(i) + 2*k3u(i) + k4u(i))/6;
v(i+1) = v(i) + (k1v(i) + 2*k2v(i) + 2*k3v(i) + k4v(i))/6;

if y(i+1) < 0
    break;
end

end

x = x(1:i+1);
y = y(1:i+1);
u = u(1:i+1);
v = v(1:i+1);
k1u = k1u(1:i+1);
k2u = k2u(1:i+1);
k3u = k3u(1:i+1);
k4u = k4u(1:i+1);
k1v = k1v(1:i+1);
k2v = k2v(1:i+1);
k3v = k3v(1:i+1);
k4v = k4v(1:i+1);
T = (i+1)*step;

disp(T)

end

```

6.2 Κώδικας adjoint

```

clc
clear
close all

format long g

xtar = 20;
r1 = 200;
r2 = 50;
rair = 1.225;
cd = 0.47 + 2/80;
g = 9.81;
ymax = 11;
ymin = 8;

t0 = 0;
tf = 10;
dt = 0.001;

x0 = 0;
y0 = 0;

psif = 0;
Ff = 0;

```

```

u0 = 8; %b1
v0 = 20; %b2
E = 5; %b3
M = 1.3;
count=1;

kill = false;
for p = 1:20 %ALM

    for k = 1:100
        R = (0.75/pi)^(1/3) * (0.5/r1 + E/r2)^(1/3);
        if imag(R) ~= 0
            kill = true;
            break;
        end
        A = pi * R^2;
        m = 0.5+E;

        F1 = @(u,v) -(2*m)^(-1)*rair*cd*A*sqrt(u^2 + v^2)*u;
        F2 = @(u,v) -g - (2*m)^(-1)*rair*cd*A*sqrt(u^2 + v^2)*v;

        [x,y,u,v,T,k1u,k1v,k2u,k2v,k3u,k3v,k4u,k4v] = RK4(F1,F2,x0,y0,u0,v0,dt,tf);
        Tp(count)=T;
        count = count+1;
        [~, index] = min(abs(x - 10));
        ywall = y(index);

        B = -(rair * cd * A)/(2*m);

        uprime = @(u,v) B * sqrt(u^2 + v^2)*u;
        vprime = @(u,v) -g + B * sqrt(u^2 + v^2) * v;
        vel = @(u,v) sqrt(u^2 + v^2);

        F3 = @(Psi,F,Psiprime,Fprime,u,v,uprime,vprime,vel) -B*(u^2/vel + vel)*Psiprime -
        (B/vel)*(-(u*uprime + v*vprime)*(u^2/vel^2) + 3*u*uprime + v*vprime)*Psi -
        (B*u*v/vel)*Fprime - (B/vel)*(-(u*uprime + v*vprime)*(u*v/vel^2) + v*uprime +
        u*vprime)*F;
        F4 = @(Psi,F,Psiprime,Fprime,u,v,uprime,vprime,vel) -B*(v^2/vel + vel)*Fprime -
        (B/vel)*(-(u*uprime + v*vprime)*(v^2/vel^2) + 3*v*vprime + u*uprime)*F -
        (B*u*v/vel)*Psiprime - (B/vel)*(-(u*uprime + v*vprime)*(u*v/vel^2) + v*uprime +
        u*vprime)*Psi;
        F5 = @(Psi,F,u,v,vel) -((rair * cd * A * vel)/(2*m^2)) * (Psi * u + F * v);
        F6 = @(Psi,F,u,v,vel) ((rair * cd * vel)/(2 * m * r2)) * (pi/6)^(1/3) * (0.5/r1 +
        E/r2)^(-1/3) * (Psi * u + F * v);

        Psiprimef = 2*M*(x(end)-xtar);
        % Fprimef = -(1+2*M*(x(end)-xtar)*u(end))/(v(end)); %Χωρίς τοίχο
        if ywall - R < ymin
            Fprimef = -(1+2*M*(x(end)-xtar)*u(end))/(v(end)) + 2 * M * (ywall - R - ymin);
        elseif ywall + R > ymax
            Fprimef = -(1+2*M*(x(end)-xtar)*u(end))/(v(end)) + 2 * M * (ywall + R - ymax);
        else
            Fprimef = -(1+2*M*(x(end)-xtar)*u(end))/(v(end));
        end
        [Psi,F,Psiprime,Fprime,Df3] =
        RK4_fae(F3,F4,psif,Ff,Psiprimef,Fprimef,dt,u,v,uprime,vprime,vel,F5,F6,k1u,k1v,k2u,k2
        v,k3u,k3v);

        Df1 = -Psi(1);
        Df2 = -F(1);

        if E - (0.25-0.012*p) * Df3 < 0

```

```

        kill = true;
        break;
    end
    E = E - (0.25-0.012*p) * Df3;
    u0 = u0 - (0.25-0.012*p) * Df1;
    v0 = v0 - (0.25-0.012*p) * Df2;
    Ep(count)=E;
    up(count)=u0;
    vp(count)=v0;
    end
    if kill
        break;
    end
    if M < 30
        M = M*1.03;
    end
end

```

```

b = [u0 v0 E]
figure(1)
plot(x,y,'LineWidth',3)
grid on
xlabel('x [m]')
ylabel('y [m]')
title('Τροχιά μάζας')

```

```

function [Psi,F,Psiprime,Fprime,Df3] =
RK4_fae(F1,F2,Psif,Ff,Psiprimef,Fprimef,step,u,v,uprime,vprime,vel,F5,F6,k1u,k1v,k2u,
k2v,k3u,k3v)

```

```

    size = length(u);
    Psi = zeros(size,1);
    Psi(size) = Psif;
    F = zeros(size,1);
    F(size) = Ff;
    Psiprime = zeros(size,1);
    Psiprime(size) = Psiprimef;
    Fprime = zeros(size,1);
    Fprime(size) = Fprimef;
    Df3 = 0;

    for i = size:-1:2

        k1psi = -step * Psiprime(i);
        k1f = -step * Fprime(i);
        k1uprime = -step*uprime(u(i),v(i));
        k1vprime = -step*vprime(u(i),v(i));
        k1psiprime = -step *
F1(Psi(i),F(i),Psiprime(i),Fprime(i),u(i),v(i),uprime(u(i),v(i)),vprime(u(i),v(i)),ve
l(u(i),v(i)));
        k1fprime = -step *
F2(Psi(i),F(i),Psiprime(i),Fprime(i),u(i),v(i),uprime(u(i),v(i)),vprime(u(i),v(i)),ve
l(u(i),v(i)));

        k2psi = -step * (Psi(i) + k1psiprime/2);
        k2f = -step * (Fprime(i) + k1fprime/2);
        k2uprime = -step*uprime(u(i)+k1uprime/2,v(i)+k1vprime/2);
        k2vprime = -step*vprime(u(i)+k1uprime/2,v(i)+k1vprime/2);
        k2psiprime = -step * F1(Psi(i) +
k1psi/2,F(i)+k1f/2,Psiprime(i)+k1psiprime/2,Fprime(i)+k1fprime/2,u(i)+k1u(i),v(i)+k1v

```

```

(i),uprime(u(i)+k1uprime/2,v(i)+k1vprime/2),vprime(u(i)+k1uprime/2,v(i)+k1vprime/2),v
el(u(i)+k1uprime/2,v(i)+k1vprime/2));
k2fprime = -step *
F2(Psi(i)+k1psi/2,F(i)+k1f/2,Psiprime(i)+k1psiprime/2,Fprime(i)+k1fprime/2,u(i)+k1u(i)
),v(i)+k1v(i),uprime(u(i)+k1uprime/2,v(i)+k1vprime/2),vprime(u(i)+k1uprime/2,v(i)+k1v
prime/2),vel(u(i)+k1uprime/2,v(i)+k1vprime/2));

k3psi = -step * (Psi(i) + k2psiprime/2);
k3f = -step * (Fprime(i) + k2fprime/2);
k3uprime = -step*uprime(u(i)+k2uprime/2,v(i)+k2vprime/2);
k3vprime = -step*vprime(u(i)+k2uprime/2,v(i)+k2vprime/2);
k3psiprime = -step * F1(Psi(i) +
k2psi/2,F(i)+k2f/2,Psiprime(i)+k2psiprime/2,Fprime(i)+k2fprime/2,u(i)+k2u(i),v(i)+k2v
(i),uprime(u(i)+k2uprime/2,v(i)+k2vprime/2),vprime(u(i)+k2uprime/2,v(i)+k2vprime/2),v
el(u(i)+k2uprime/2,v(i)+k2vprime/2));
k3fprime = -step *
F2(Psi(i)+k2psi/2,F(i)+k2f/2,Psiprime(i)+k2psiprime/2,Fprime(i)+k2fprime/2,u(i)+k2u(i)
),v(i)+k2v(i),uprime(u(i)+k2uprime/2,v(i)+k2vprime/2),vprime(u(i)+k2uprime/2,v(i)+k2v
prime/2),vel(u(i)+k2uprime/2,v(i)+k2vprime/2));

k4psi = -step * (Psi(i) + k3psiprime);
k4f = -step * (Fprime(i) + k3fprime);
k4psiprime = -step * F1(Psi(i) +
k3psi,F(i)+k3f,Psiprime(i)+k3psiprime,Fprime(i)+k3fprime,u(i)+k3u(i),v(i)+k3v(i),upri
me(u(i)+k3uprime/2,v(i)+k3vprime/2),vprime(u(i)+k3uprime/2,v(i)+k3vprime/2),vel(u(i)+
k3uprime/2,v(i)+k3vprime/2));
k4fprime = -step *
F2(Psi(i)+k3psi,F(i)+k3f,Psiprime(i)+k3psiprime,Fprime(i)+k3fprime,u(i)+k3u(i),v(i)+k
3v(i),uprime(u(i)+k3uprime/2,v(i)+k3vprime/2),vprime(u(i)+k3uprime/2,v(i)+k3vprime/2)
,vel(u(i)+k3uprime/2,v(i)+k3vprime/2));

Psi(i-1) = Psi(i) + (k1psi + 2 * k2psi + 2 * k3psi + k4psi)/6;
F(i-1) = F(i) + (k1f + 2 * k2f + 2 * k3f + k4f)/6;
Psiprime(i-1) = Psiprime(i) + (k1psiprime + 2 * k2psiprime + 2 * k3psiprime +
k4psiprime)/6;
Fprime(i-1) = Fprime(i) + (k1fprime + 2 * k2fprime + 2 * k3fprime +
k4fprime)/6;

Df3 = Df3 + step * (F5(Psi(i),F(i),u(i),v(i),vel(u(i),v(i))) + F5(Psi(i-
1),F(i-1),u(i-1),v(i-1),vel(u(i-1),v(i-1)))))/2 + step *
(F6(Psi(i),F(i),u(i),v(i),vel(u(i),v(i))) + F6(Psi(i-1),F(i-1),u(i-1),v(i-1),vel(u(i-
1),v(i-1)))))/2;
end

%disp(Df3)

end

function [x,y,u,v,T,k1u,k1v,k2u,k2v,k3u,k3v,k4u,k4v] = RK4(F1,F2,x0,y0,u0,v0,step,tf)

x = zeros(tf/step,1);
x(1) = x0;
y = zeros(tf/step,1);
y(1) = y0;
u = zeros(tf/step,1);
u(1) = u0; %b1
v = zeros(tf/step,1);
v(1) = v0; %b2
k1x = zeros(tf/step,1);
k2x = zeros(tf/step,1);
k3x = zeros(tf/step,1);
k4x = zeros(tf/step,1);

```



```

k1y = zeros(tf/step,1);
k2y = zeros(tf/step,1);
k3y = zeros(tf/step,1);
k4y = zeros(tf/step,1);

k1u = zeros(tf/step,1);
k2u = zeros(tf/step,1);
k3u = zeros(tf/step,1);
k4u = zeros(tf/step,1);

k1v = zeros(tf/step,1);
k2v = zeros(tf/step,1);
k3v = zeros(tf/step,1);
k4v = zeros(tf/step,1);

for i = 1:(tf/step - 1)

    k1x(i) = step * u(i);
    k1y(i) = step * v(i);
    k1u(i) = step * F1(u(i),v(i));
    k1v(i) = step * F2(u(i),v(i));

    k2x(i) = step * (u(i) + k1u(i)/2);
    k2y(i) = step * (v(i) + k1v(i)/2);
    k2u(i) = step * F1(u(i) + k1u(i)/2,v(i) + k1v(i)/2);
    k2v(i) = step * F2(u(i) + k1u(i)/2,v(i) + k1v(i)/2);

    k3x(i) = step * (u(i) + k2u(i)/2);
    k3y(i) = step * (v(i) + k2v(i)/2);
    k3u(i) = step * F1(u(i) + k2u(i)/2,v(i) + k2v(i)/2);
    k3v(i) = step * F2(u(i) + k2u(i)/2,v(i) + k2v(i)/2);

    k4x(i) = step * (u(i) + k3u(i));
    k4y(i) = step * (v(i) + k3v(i));
    k4u(i) = step * F1(u(i) + k3u(i)/2,v(i) + k3v(i)/2);
    k4v(i) = step * F2(u(i) + k3u(i)/2,v(i) + k3v(i)/2);

    x(i+1) = x(i) + (k1x(i) + 2*k2x(i) + 2*k3x(i) + k4x(i))/6;
    y(i+1) = y(i) + (k1y(i) + 2*k2y(i) + 2*k3y(i) + k4y(i))/6;
    u(i+1) = u(i) + (k1u(i) + 2*k2u(i) + 2*k3u(i) + k4u(i))/6;
    v(i+1) = v(i) + (k1v(i) + 2*k2v(i) + 2*k3v(i) + k4v(i))/6;

    if y(i+1) < 0
        break;
    end

end

x = x(1:i+1);
y = y(1:i+1);
u = u(1:i+1);
v = v(1:i+1);
k1u = k1u(1:i+1);
k2u = k2u(1:i+1);
k3u = k3u(1:i+1);
k4u = k4u(1:i+1);
k1v = k1v(1:i+1);
k2v = k2v(1:i+1);
k3v = k3v(1:i+1);
k4v = k4v(1:i+1);
T = (i+1)*step;

```

```
disp(T)
```

```
end
```

6.3 Κώδικας CV

```
clc
clear
close all

format long g

xtar = 20;
r1 = 200;
r2 = 50;
rair = 1.225;
cd = 0.47 + 2/80;
g = 9.81;

t0 = 0;
tf = 10;
dt = 0.001;

x0 = 0;
y0 = 0;

psif = 0;
Ff = 0;

u0 = 8; %b1
v0 = 20; %b2
E = 0.1; %b3
M = 1.3;
epsilon = 10^(-5);

kill = false;
for p = 1:20 %ALM

    for k = 1:500
        R = (0.75/pi)^(1/3) * (0.5/r1 + E/r2)^(1/3);
        if imag(R) ~= 0
            kill = true;
            break;
        end
        A = pi * R^2;
        m = 0.5+E;

        F1 = @(vx,vy) -(2*m)^(-1)*rair*cd*A*sqrt(vx^2 + vy^2)*vx;
        F2 = @(vx,vy) -g - (2*m)^(-1)*rair*cd*A*sqrt(vx^2 + vy^2)*vy;

        [x,y,u,v,T] = RK4(F1,F2,x0,y0,u0,v0,dt,tf);

        [Df1,Df2] = cvm(F1,F2,x0,y0,u0,v0,dt,tf,epsilon,M,R);

        E = complex(E,epsilon);
```

```

R = (0.75/pi)^(1/3) * (0.5/r1 + E/r2)^(1/3);
A = pi * R^2;
m = 0.5+E;
F1 = @(vx,vy) -(2*m)^(-1)*rair*cd*A*sqrt(vx^2 + vy^2)*vx;
F2 = @(vx,vy) -g - (2*m)^(-1)*rair*cd*A*sqrt(vx^2 + vy^2)*vy;

[x5,y5,~,~,T5] = RK4(F1,F2,x0,y0,u0,v0,dt,tf);

 [~, index] = min(abs(x5 - 10));
 ywall = y5(index);

if ywall - R < 8
    Df3 = imag((T5 + M*((x5(end)-xtar)^2) + M*(ywall-R-8)^2 + y5(end)))/(epsilon);
elseif ywall + R > 11
    Df3 = imag((T5 + M*((x5(end)-xtar)^2) + M*(ywall+R-11)^2 + y5(end)))/(epsilon);
else
    Df3 = imag((T5 + M*((x5(end)-xtar)^2) + y5(end)))/(epsilon);
end
fprintf('x(T) = %f + %fi p = %d k = %d \n',real(x(end)),imag(x(end)),p,k)
E = real(E);

if E - 0.01 * Df3 < 0
    kill = true;
    break;
end
u0 = u0 - 0.01 * Df1;
v0 = v0 - 0.01 * Df2;
E = E - 0.01 * Df3;
end
if kill
    break;
end
if M < 30
    M = M*1.02;
end
end

function [df1,df2] = cvm(F1,F2,x0,y0,u0,v0,step,tf,epsilon,M,R)
    df1 = 0;
    df2 = 0;

    [x1,y1,~,~,T1] = RK4(F1,F2,x0,y0,complex(u0,epsilon),v0,step,tf);
    [~, index1] = min(abs(x1 - 10));
    ywall = y1(index1);

    if ywall - R < 8
        df1 = imag((T1 + M*((x1(end)-20)^2) + M*(ywall-R-8)^2 + y1(end)))/(epsilon);
    elseif ywall + R > 11
        df1 = imag((T1 + M*((x1(end)-20)^2) + M*(ywall+R-11)^2 + y1(end)))/(epsilon);
    else
        df1 = imag((T1 + M*((x1(end)-20)^2) + y1(end)))/(epsilon);
    end

    [x2,y2,~,~,T2] = RK4(F1,F2,x0,y0,u0,complex(v0,epsilon),step,tf);
    [~, index2] = min(abs(x2 - 10));
    ywall = y2(index2);

    if ywall - R < 8
        df2 = imag((T2 + M*((x2(end)-20)^2) + M*(ywall-R-8)^2 + y2(end)))/(epsilon);
    elseif ywall + R > 11

```

```

        df2 = imag((T2 + M*((x2(end)-20)^2) + M*(ywall+R-11)^2 + y2(end)))/(epsilon);
    else
        df2 = imag((T2 + M*((x2(end)-20)^2) + y2(end)))/(epsilon);
    end

end

function [x,y,u,v,T,k1u,k1v,k2u,k2v,k3u,k3v,k4u,k4v] = RK4(F1,F2,x0,y0,u0,v0,step,tf)

x = zeros(tf/step,1);
x(1) = x0;
y = zeros(tf/step,1);
y(1) = y0;
u = zeros(tf/step,1);
u(1) = u0; %b1
v = zeros(tf/step,1);
v(1) = v0; %b2
k1x = zeros(tf/step,1);
k2x = zeros(tf/step,1);
k3x = zeros(tf/step,1);
k4x = zeros(tf/step,1);

k1y = zeros(tf/step,1);
k2y = zeros(tf/step,1);
k3y = zeros(tf/step,1);
k4y = zeros(tf/step,1);

k1u = zeros(tf/step,1);
k2u = zeros(tf/step,1);
k3u = zeros(tf/step,1);
k4u = zeros(tf/step,1);

k1v = zeros(tf/step,1);
k2v = zeros(tf/step,1);
k3v = zeros(tf/step,1);
k4v = zeros(tf/step,1);

for i = 1:(tf/step - 1)

    k1x(i) = step * u(i);
    k1y(i) = step * v(i);
    k1u(i) = step * F1(u(i),v(i));
    k1v(i) = step * F2(u(i),v(i));

    k2x(i) = step * (u(i) + k1u(i)/2);
    k2y(i) = step * (v(i) + k1v(i)/2);
    k2u(i) = step * F1(u(i) + k1u(i)/2,v(i) + k1v(i)/2);
    k2v(i) = step * F2(u(i) + k1u(i)/2,v(i) + k1v(i)/2);

    k3x(i) = step * (u(i) + k2u(i)/2);
    k3y(i) = step * (v(i) + k2v(i)/2);
    k3u(i) = step * F1(u(i) + k2u(i)/2,v(i) + k2v(i)/2);
    k3v(i) = step * F2(u(i) + k2u(i)/2,v(i) + k2v(i)/2);

    k4x(i) = step * (u(i) + k3u(i));
    k4y(i) = step * (v(i) + k3v(i));
    k4u(i) = step * F1(u(i) + k3u(i)/2,v(i) + k3v(i)/2);
    k4v(i) = step * F2(u(i) + k3u(i)/2,v(i) + k3v(i)/2);

    x(i+1) = x(i) + (k1x(i) + 2*k2x(i) + 2*k3x(i) + k4x(i))/6;
    y(i+1) = y(i) + (k1y(i) + 2*k2y(i) + 2*k3y(i) + k4y(i))/6;

```

```

u(i+1) = u(i) + (k1u(i) + 2*k2u(i) + 2*k3u(i) + k4u(i))/6;
v(i+1) = v(i) + (k1v(i) + 2*k2v(i) + 2*k3v(i) + k4v(i))/6;

    if y(i+1) < 0
        break;
    end

end

x = x(1:i+1);
y = y(1:i+1);
u = u(1:i+1);
v = v(1:i+1);
k1u = k1u(1:i+1);
k2u = k2u(1:i+1);
k3u = k3u(1:i+1);
k4u = k4u(1:i+1);
k1v = k1v(1:i+1);
k2v = k2v(1:i+1);
k3v = k3v(1:i+1);
k4v = k4v(1:i+1);
T = (i+1)*step;

% disp(T)

end

```