



Πανεπιστήμιο Ιωαννίνων

Τμήμα Μηχανικών Η/Υ και Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών με τίτλο:

“Μηχανική Δεδομένων και Υπολογιστικών Συστημάτων”

## **1<sup>η</sup> Εργασία στο Μεταπτυχιακό μάθημα της Βελτιστοποίησης**

Υπεύθυνος Καθηγητής:

Παρσόπουλος Κωνσταντίνος

Φοιτητές:

Βαρδάκας Γεώργιος Α.Μ.432

Τριανταλή Δήμητρα Α.Μ.431

Ιωάννινα 2020

## Περιεχόμενα

Κεφάλαιο	Τίτλος
1	Στόχος της εργασίας
2	Δεδομένα
	Υλοποίηση
	3.1 Βιβλιοθήκες
	Συναρτήσεις
3	3.2.1 Συναρτήσεις για τα δεδομένα
	3.2 3.2.2 Συναρτήσεις για τις μεθόδους βελτιστοποίησης
	3.2.3 Συναρτήσεις για την πρόβλεψη των κρουσμάτων
	3.2.4 Η κύρια συνάρτηση
	Αποτελέσματα
	4.1 Αποτελέσματα για την πρώτη περίπτωση
	4.2 Αποτελέσματα για τη δεύτερη περίπτωση
4	4.3 Αποτελέσματα για την τρίτη περίπτωση
	4.4 Αποτελέσματα για την τέταρτη περίπτωση
	4.5 Αποτελέσματα για την πέμπτη περίπτωση
	4.6 Σύνοψη αποτελεσμάτων
	Προβλέψεις
	5.1 Προβλέψεις για την πρώτη περίπτωση
	5.2 Προβλέψεις για τη δεύτερη περίπτωση
5	5.3 Προβλέψεις για την τρίτη περίπτωση
	5.4 Προβλέψεις για την τέταρτη περίπτωση
	5.5 Προβλέψεις για την πέμπτη περίπτωση
	5.6 Σύνοψη των προβλέψεων
	Παράρτημα
6	Σύγκριση των 4 μεθόδων βελτιστοποίησης ως προς τον χρόνο σε δευτερόλεπτα που απαίτησαν

## Κεφάλαιο 1

### Στόχος της εργασίας

Ο στόχος της παρούσας εργασίας είναι να υπολογιστεί ένα πολυωνμικό μοντέλο 4ου βαθμού της μορφής  $p(a, x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4$ ,

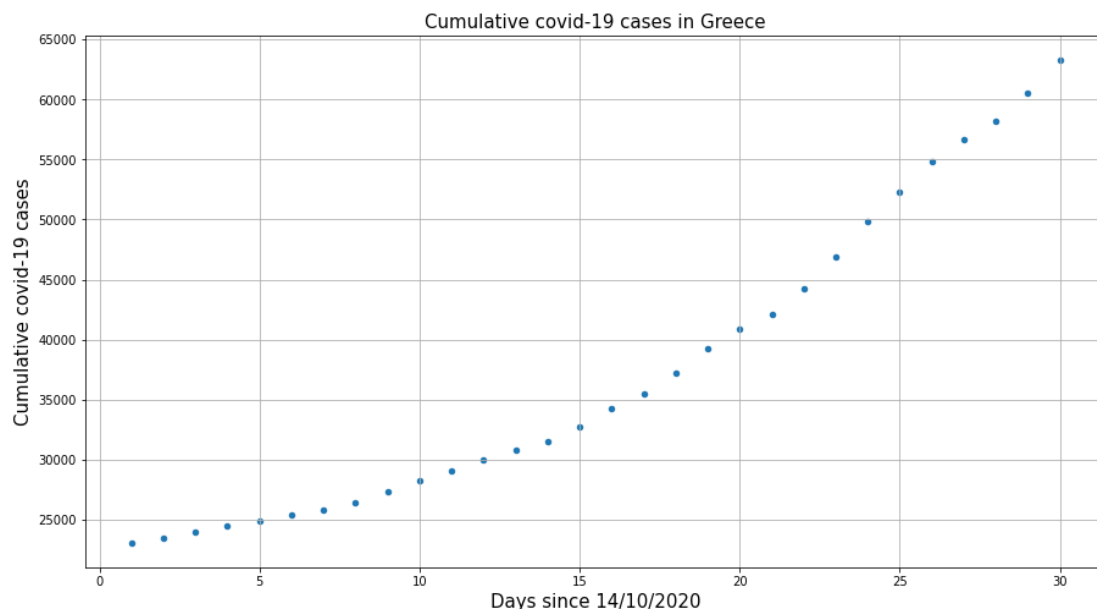
όπου  $a = (a_0, a_1, a_2, a_3, a_4)^T$  το άγνωστο διάνυσμα των συντελεστών του μοντέλου, το οποίο περιγράφει τα δεδομένα με τον καλύτερο δυνατό τρόπο, δηλαδή ελαχιστοποιεί τη συνάρτηση τετραγωνικού σφάλματος  $f(a) = \sum_{i=1}^M [p(a, x_i) - y_i]^2$ , όπου  $M=30$  το πλήθος των διαθέσιμων μετρήσεων στα δεδομένα.

## Κεφάλαιο 2

### Δεδομένα

Η εργασία στηρίχθηκε στο αρχείο 'covid\_data\_30\_GR.dat' το οποίο μας ανατέθηκε. Στο αρχείο δίνεται ο συσσωρευτικός αριθμός κρουσμάτων του covid-19 στην Ελλάδα για διάστημα 30 ημερών από 14 Οκτωβρίου 2020 μέχρι και 12 Νοεμβρίου 2020. Στην πρώτη στήλη των δεδομένων δίνεται ο αριθμός  $x_i$  της ημέρας ( $i=1,2,\dots,30$ ) ξεκινώντας από την 14η Οκτωβρίου (ημέρα 1) και στη δεύτερη στήλη δίνεται ο αντίστοιχος συνολικός αριθμός κρουσμάτων  $y_i$  που έχει καταγραφεί στη χώρα από την αρχή της πανδημίας (1 Μαρτίου 2020) μέχρι την αντίστοιχη ημέρα (π.χ. μέχρι την ημέρα 1, δηλαδή από την 1η Μαρτίου 2020 μέχρι τις 14 Οκτωβρίου 2020 είχαν καταγραφεί 23060 κρούσματα).

Στην επόμενη γραφική παράσταση απεικονίζονται τα κρούσματα που καταγράφηκαν στην Ελλάδα σε συνάρτηση με τις αντίστοιχες ημέρες. Παρατηρούμε ότι η γραφική παράσταση είναι ανοδική, κάτι το οποίο είναι αναμενόμενο μιας και καθημερινά υπάρχει τουλάχιστον 1 κρούσμα, δηλαδή ο συσσωρευτικός αριθμός των κρουσμάτων συνεχώς αυξάνεται.



## Κεφάλαιο 3

### Υλοποίηση

#### 3.1.Βιβλιοθήκες

Για την υλοποίηση των βασικών συναρτήσεων της εργασίας χρησιμοποιήσαμε τις βιβλιοθήκες numpy και pandas. Για τον υπολογισμό του χρόνου που απαιτήθηκε για να δοθεί η λύση κάναμε χρήση της βιβλιοθήκης time. Για τις γραφικές αναπαραστάσεις χρησιμοποιήσαμε τη βιβλιοθήκη matplotlib. Επίσης, χρησιμοποιήσαμε τη βιβλιοθήκη sys για να τρέχει το πρόγραμμά μας μέσω του τερματικού.

#### 3.2.Συναρτήσεις

##### 3.2.1. Συναρτήσεις για τα δεδομένα

Αρχικά, δημιουργήσαμε ένα dataframe με τα δεδομένα του αρχείου 'covid\_data\_30\_GR.dat'. Ονομάσαμε την πρώτη στήλη 'Day' και τη δεύτερη στήλη 'Cases'. Παρατηρήσαμε ότι τα δεδομένα έχουν σημαντικά διαφορετική κλίμακα και για αυτό δημιουργήσαμε 2 νέες στήλες. Στη στήλη 'Scaled Day' τοποθετήσαμε τα δεδομένα της στήλης 'Day' διαιρεμένα με το 10. Στη στήλη 'Scaled Cases' τοποθετήσαμε τα δεδομένα της στήλης 'Cases' διαιρεμένα με το 10000. Παρατηρήσαμε τότε ότι τα δεδομένα των 2 νέων στηλών είναι πλέον της ίδιας κλίμακας.

Στη συνέχεια, προσδιορίσαμε το μοντέλο μας. Η συνάρτηση model(a, x) επιστρέφει το μοντέλο μας  $p(a, x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4$ , έχοντας πάρει ως είσοδο το  $a = (a_0, a_1, a_2, a_3, a_4)^T$ , δηλαδή το άγνωστο διάνυσμα των συντελεστών του μοντέλου που θέλουμε να προσδιορίσουμε και το δοθέν σημείο x. Υπολογίσαμε ακόμη με τη συνάρτηση model\_gradient(a, x) το διάνυσμα κλίσης του μοντέλου:  $\frac{dp(a, x)}{da} = (1 \ x \ x^2 \ x^3 \ x^4)^T$  και με την model\_hessian(a, x) τον εσσιανό πίνακα του μοντέλου:  $\frac{d^2p(a, x)}{da^2}$  που είναι ο μηδενικός  $5 \times 5$  πίνακας. Η συνάρτηση του μοντέλου είναι συνεχώς παραγωγίσιμη ως πολυωνυμική συνάρτηση.

Έπειτα, προσδιορίσαμε τη συνάρτηση σφάλματος, την οποία θέλουμε να ελαχιστοποιήσουμε. Η συνάρτηση error\_function(model\_parameters) επιστρέφει το συνολικό σφάλμα  $f(a) = \sum_{i=1}^{30} [p(a, x_i) - y_i]^2$ , έχοντας πάρει ως είσοδο τις παραμέτρους του μοντέλου που επιλέχθηκε. Υπολογίσαμε ακόμη με τη συνάρτηση gradient\_of\_error\_function(model\_parameters) την κλίση της συνάρτησης σφάλματος:  $\frac{df(a)}{da} = 2 \sum_{i=1}^{30} [p(a, x_i) - y_i] \frac{dp(a, x_i)}{da}$  και με τη hessian\_of\_error\_function(model\_parameters) τον εσσιανό πίνακα της συνάρτησης σφάλματος:  $\frac{d^2f(a)}{da^2} = 2 \sum_{i=1}^{30} \left\{ \frac{dp(a, x_i)}{da} * \left( \frac{dp(a, x_i)}{da} \right)^T + [p(a, x_i) - y_i] \frac{d^2p(a, x)}{da^2} \right\}$ . Η συνάρτηση σφάλματος είναι συνεχώς παραγωγίσιμη ως σύνθεση συνεχών και παραγωγίσιμων συναρτήσεων.

##### 3.2.2.Συναρτήσεις για τις μεθόδους βελτιστοποίησης

Ακολούθησε η υλοποίηση των 4 μεθόδων για την ελαχιστοποίηση της συνάρτησης σφάλματος. Και στις 4 μεθόδους θέσαμε δύο κριτήρια τερματισμού. Το πρώτο ήταν να γίνουν μέχρι το πολύ 1000 επαναλήψεις για την εύρεση του ελαχιστοποιητή. Το δεύτερο κριτήριο ήταν η νόρμα της κλίσης της συνάρτησης σφάλματος να είναι το πολύ ίση με  $10^{-6}$ . Για την υλοποίηση των 4 μεθόδων δημιουργήσαμε τις ακόλουθες βοηθητικές συναρτήσεις.

- Η `cholesky_method(hessian)` ελέγχει τη μέθοδο Cholesky. Παίρνει ως είσοδο έναν πίνακα και αν ο πίνακας είναι θετικά ορισμένος επιστρέφει True, ενώ αν ο πίνακας δεν είναι θετικά ορισμένος δημιουργείται λάθος και επιστρέφεται False.
- Η `hessian_modification(hessian_xk, beta)` τροποποιεί έναν πίνακα στηριζόμενη στην ιδιότητα ότι ένας πίνακας με αυστηρή διαγώνια κυριαρχία και θετικά διαγώνια στοιχεία είναι θετικά ορισμένος. Παίρνει ως είσοδο έναν πίνακα και μια σταθερά beta. Η συνάρτηση βρίσκει τα διαγώνια στοιχεία του πίνακα και το ελάχιστο τους. Αν το ελάχιστο είναι αρνητικός αριθμός, τότε παίρνουμε τον αντίθετο του, τον αυξάνουμε κατά την παράμετρο beta και τον θέτουμε στην παράμετρο t. Καλούμε τη συνάρτηση `cholesky_method()` με παράμετρο τον πίνακα έχοντας αυξήσει τα διαγώνια στοιχεία του κατά t. Αν η συνάρτηση `cholesky_method()` δημιουργεί λάθος, τότε παίρνουμε ως t το μέγιστο ανάμεσα στα  $t^2, \beta$ . Θέσαμε ως περιορισμό η συνάρτηση να κάνει το πολύ 100 επαναλήψεις. Η συνάρτηση επιστρέφει τον τροποποιημένο πίνακα L ή το μοναδιαίο πίνακα σε περίπτωση που εξαντληθούν οι 100 επαναλήψεις.
- Η `strong_wolf_conditions(pk, xk, a_max, c2)` ελέγχει τις ισχυρές συνθήκες Wolfe. Παίρνει ως είσοδο τη διεύθυνση pk, το σημείο xk, τη μέγιστη τιμή του βήματος και την παράμετρο της συνθήκης Armijo. Το μικρότερο βήμα το θέσαμε να ναι ίσο με 0, ενώ ως πρώτο βήμα α επιλέξαμε να καλέσουμε τη συνάρτηση `interpolate()` στο [0,2]. Αρχικά, κατασκευάσαμε τη συνάρτηση  $\varphi(\alpha) = f(xk + \alpha pk)$  και στη συνέχεια ξεκινήσαμε τον έλεγχο των ισχυρών κριτηρίων Wolfe. Αν το α ικανοποιεί και τα 2 κριτήρια, τότε είναι το επιθυμητό. Αν το α δεν ικανοποιεί τη συνθήκη Armijo καλούμε τη `zoom()` στο [0,α] και παίρνουμε νέο α. Αν το α ικανοποιεί τη συνθήκη Armijo, αλλά δεν ικανοποιεί τη συνθήκη καμπυλότητας υπολογίζουμε την παράγωγο της συνάρτησης  $\varphi(\alpha)$ . Αν η παράγωγος είναι μη αρνητική σημαίνει ότι οι ελαχιστοποιητές βρίσκονται αριστερότερα της περιοχής που πέσαμε, άρα είμαστε σε ανοδική κλίση και καλούμε τη `zoom()` στο [0,α] και λαμβάνουμε νέο α. Αν η παράγωγος είναι αρνητική σημαίνει ότι οι ελαχιστοποιητές βρίσκονται δεξιότερα της περιοχής που πέσαμε, άρα είμαστε σε καθοδική κλίση και παίρνουμε νέο α στο (α,2).
- Η `interpolate(a_low, a_high)` κάνει παρεμβολή στο διάστημα [a\_low, a\_high]. Ως μέθοδο παρεμβολής επιλέξαμε τη μέθοδο της διχοτόμησης, δηλαδή η κλήση της συνάρτησης επιστρέφει το μέσο του διαστήματος που δόθηκε.
- Η `zoom(c1, c2, line, f_0, f_gradient_0, pk, xk, a_low, a_high)` είναι υπεύθυνη να κάνει παρεμβολή και να ελέγχει τις συνθήκες wolfe. Παίρνει ως είσοδο τις τιμές των παραμέτρων των συνθηκών wolfe, τη συνάρτηση  $\varphi(\alpha)$ , τη συνάρτηση  $\varphi(\alpha)$  για  $\alpha_{min}$ , την παράγωγο της  $\varphi(\alpha)$  για  $\alpha_{min}$ , τη διεύθυνση pk, το σημείο xk και το δοθέν διάστημα  $[\alpha_{low}, \alpha_{high}]$ . Αρχικά, καλούμε τη συνάρτηση `interpolate()` στο δοθέν διάστημα για την επιλογή ενός α και υπολογίζουμε τη συνάρτηση  $\varphi(\alpha) = f(xk + \alpha pk)$ . Αν ικανοποιείται η συνθήκη Armijo, τότε το νέο μας διάστημα είναι το  $[\alpha_{low}, \alpha]$ . Αλλιώς, υπολογίζουμε την παράγωγο της  $\varphi(\alpha)$  και αν ισχύει η συνθήκη καμπυλότητας επιστρέφουμε ως βήμα το α. Αν η παράγωγος της  $\varphi(\alpha)$  επί το μήκος του διαστήματος είναι μη αρνητικός αριθμός, τότε μειώνουμε το μέγιστο α. Θέσαμε ως περιορισμό η συνάρτηση να κάνει το πολύ 100 επαναλήψεις.
- Η `get_direction(xk, Bk, deltak)` επιλέγει μια διεύθυνση μείωσης εντός της ασφαλούς περιοχής που επιφέρει επαρκή μείωση στην τιμή του μοντέλου. Η συνάρτηση παίρνει ως είσοδο το σημείο xk, τον εσσιανό πίνακα της συνάρτησης σφάλματος και την ακτίνα της ασφαλούς περιοχής Δk. Αρχικά, η συνάρτηση υπολογίζει το Newton point  $p^B = -\text{hessian}B_k^{-1} * \text{gradient}(xk)$ . Αν το σημείο αυτό πέφτει εντός της ασφαλούς περιοχής τότε βρήκαμε το επιθυμητό σημείο. Αλλιώς υπολογίζουμε το Cauchy point  $p^u = -\frac{\text{gradient}(xk)^T * \text{gradient}(xk)}{\text{gradient}(xk)^T * B_k * \text{gradient}(xk)} * \text{gradient}(xk)$ . Αν το Cauchy point είναι εκτός της ασφαλούς περιοχής, το κόβουμε στο σύνορο και επιστρέφουμε το  $p^* = -\frac{\text{gradient}(xk)}{\|\text{gradient}(xk)\|} \Delta k$ . Αν το Cauchy point είναι εντός της ασφαλούς περιοχής,

εφαρμόζουμε τη μέθοδο dogleg. Καλώντας τη συνάρτηση solve() υπολογίζουμε το  $t$  και επιστρέφουμε το  $p^* = p^u + (t^* - 1)(p^B - p^u)$ .

- Η solve(pB, pU, deltak) εφαρμόζει τη μέθοδο της διχοτόμησης παίρνοντας ως είσοδο τα Newton και Cauchy points μαζί με την ακτίνα της ασφαλούς περιοχής  $\Delta k$ . Αρχικά, η συνάρτηση θέτει ως εξίσωση προς επίλυση την  $\|p^u + (t - 1)(p^B - p^u)\|^2 - \Delta k^2$  για  $t \in [1, 2]$ . Υπολογίζει το μέσο του διαστήματος που δόθηκε. Αν η απόλυτη τιμή της εξίσωσης στο μέσο είναι μικρότερη ή ίση του σφάλματος τότε το μέσο είναι το επιθυμητό  $t$ . Αλλιώς αν η τιμή της εξίσωσης στο κάτω άκρο επί την τιμή της στο μέσο είναι αρνητική, τότε εφαρμόζω ξανά τη μέθοδο στο διάστημα ανάμεσα στο μέσο και το άνω όριο. Σε αντίθετη περίπτωση εφαρμόζω ξανά τη μέθοδο στο διάστημα ανάμεσα στο κάτω όριο και το μέσο. Αν το μήκος του δοθέντος διαστήματος είναι μικρότερο του σφάλματος, τότε το μέσο του διαστήματος είναι το επιθυμητό  $t$ .

Η πρώτη μέθοδος ελαχιστοποίησης της συνάρτησης σφάλματος ήταν η μέθοδος Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Η συνάρτηση newton\_direction\_wolf\_conditions(init\_parameters) παίρνει ως είσοδο ένα αρχικό διάνυσμα των συντελεστών του μοντέλου. Αρχικά, ελέγχει αν ο εσσιανός πίνακας είναι θετικά ορισμένος καλώντας τη βοηθητική συνάρτηση cholesky\_method(). Σε περίπτωση που δεν είναι θετικά ορισμένος τον μετασχηματίζει κατάλληλα, καλώντας τη βοηθητική συνάρτηση hessian\_modification(). Επιλέξαμε ως διεύθυνση τη διεύθυνση  $p_k = -\text{gradient}(f(x_k)) * (\text{hessian}(f(x_k)))^{-1}$ , πάνω στην οποία κάναμε ένα βήμα που ικανοποιεί τις ισχυρές συνθήκες wolfe, καλώντας τη βοηθητική συνάρτηση strong\_wolf\_conditions(). Τέλος ενημερώσαμε το επαναληπτικό σχήμα της μεθόδου ( $x_{k+1} = x_k + a_k p_k$ ).

Η δεύτερη μέθοδος ελαχιστοποίησης της συνάρτησης σφάλματος ήταν η μέθοδος Newton ασφαλούς περιοχής. Η συνάρτηση newton\_method\_safe\_region(init\_parameters) παίρνει ως είσοδο ένα αρχικό διάνυσμα των συντελεστών του μοντέλου. Αρχικά, ελέγχει αν ο εσσιανός πίνακας είναι θετικά ορισμένος καλώντας τη βοηθητική συνάρτηση cholesky\_method(). Σε περίπτωση που δεν είναι θετικά ορισμένος τον μετασχηματίζει κατάλληλα, καλώντας τη βοηθητική συνάρτηση hessian\_modification(). Η διεύθυνση επιλέχθηκε καλώντας τη βοηθητική συνάρτηση get\_direction(). Στη συνέχεια, η συνάρτηση υπολόγισε το λόγο ελέγχου της ποιότητας της ακτίνας  $p_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)}$ . Αν ο λόγος είναι κοντά στο 0, τότε στην επόμενη επανάληψη μειώνουμε την ακτίνα. Αν ο λόγος είναι κοντά στο 1 αυξάνουμε την ακτίνα στην επόμενη επανάληψη, χωρίς όμως να ξεπεράσουμε τη μέγιστη ακτίνα που θέσαμε ίση με 1. Αν ο λόγος είναι ανάμεσα στα 0 και 1, αλλά όχι κοντά στο 0, τότε η ακτίνα δε μεταβάλλεται. Αν ο λόγος είναι μεγαλύτερος του κατωφλίου  $h_{\text{tta}}$  που το θέσαμε ίσο με 0.25, τότε κάνω αποδεκτή τη διεύθυνση και ενημερώνω το επαναληπτικό σχήμα  $x_{k+1} = x_k + p_k$ , αλλιώς η διεύθυνση απορρίπτεται και θέτω  $x_{k+1} = x_k$ .

Η τρίτη μέθοδος ελαχιστοποίησης της συνάρτησης σφάλματος ήταν η μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Η συνάρτηση BFGS\_wolf\_conditions(init\_parameters, hessian\_approx) παίρνει ως είσοδο ένα αρχικό διάνυσμα των συντελεστών του μοντέλου και μία προσέγγιση του εσσιανού πίνακα. Αρχικά, μόνο για την πρώτη επανάληψη ελέγχει αν η προσέγγιση του εσσιανού πίνακα είναι θετικά ορισμένη καλώντας τη βοηθητική συνάρτηση cholesky\_method(). Σε περίπτωση που δεν είναι θετικά ορισμένη την μετασχηματίζει κατάλληλα, καλώντας τη βοηθητική συνάρτηση hessian\_modification(). Ο αλγόριθμος ελέγχει την ιδιότητα του θετικά ορισμένου πίνακα μόνο στην πρώτη επανάληψη, καθώς η μέθοδος BFGS μας εξασφαλίζει ότι διατηρείται η ιδιότητα του θετικά ορισμένου πίνακα. Εμείς διαλέξαμε ως προσέγγιση του εσσιανού το μοναδιαίο πίνακα, που είναι θετικά ορισμένος πίνακας. Επιλέξαμε ως διεύθυνση τη διεύθυνση 'Newton'  $p_k = -\text{gradient}(f(x_k)) * (\text{hessian}(f(x_k)))^{-1}$  πάνω στην οποία κάναμε ένα βήμα που ικανοποιεί τις ισχυρές συνθήκες wolfe και ενημερώσαμε το επαναληπτικό σχήμα της μεθόδου. Στη συνέχεια, ενημερώσαμε την προσέγγιση του εσσιανού

πίνακα με βάση τον τύπο των BFGS,  $H_{k+1} = (I - p_k s_k * y_k^T) * H_k * (I - p_k y_k * s_k^T) + p_k s_k * s_k^T$ .

Η τέταρτη μέθοδος ελαχιστοποίησης της συνάρτησης σφάλματος ήταν η μέθοδος steepest descent με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Η συνάρτηση `steepest_descent_wolf_conditions(init_parameters)` παίρνει ως είσοδο ένα αρχικό διάνυσμα των συντελεστών του μοντέλου. Επιλέξαμε ως διεύθυνση τη διεύθυνση απότομης καθόδου  $p_k = -\text{gradient}(f(x_k))$ , πάνω στην οποία κάναμε ένα βήμα που ικανοποιεί τις ισχυρές συνθήκες wolfe, καλώντας τη βοηθητική συνάρτηση `strong_wolf_conditions()`. Τέλος ενημερώσαμε το επαναληπτικό σχήμα της μεθόδου ( $x_{k+1} = x_k + a_k p_k$ ).

### 3.2.3. Συναρτήσεις για την πρόβλεψη των κρουσμάτων

Για το δεύτερο μέρος της εργασίας που αφορούσε τη χρήση του βέλτιστου μοντέλου για την πρόβλεψη των επόμενων κρουσμάτων αρχικά δημιουργήσαμε ένα dataframe με τα δεδομένα που βρήκαμε στη σελίδα '<https://covid19.who.int/region/euro/country/gr>'. Ονομάσαμε την πρώτη στήλη 'Day' και τις δώσαμε τις τιμές 31,32,...35 και τη δεύτερη στήλη 'Cases' και της δώσαμε τις τιμές του συσσωρευτικού αριθμού των κρουσμάτων που εντοπίστηκαν τις αντίστοιχες ημέρες στην Ελλάδα. Στη συνέχεια υλοποιήθηκαν οι επόμενες συναρτήσεις.

Η `optimizer(method_choice, init_parameters)` επιλέγει τη μέθοδο βελτιστοποίησης που θα εκτελεστεί, καθώς και με ποιο αρχικό διάνυσμα παραμέτρων. Παίρνει ως είσοδο τη μέθοδο που επιλέχθηκε και το αρχικό διάνυσμα των συντελεστών του μοντέλου που διάλεξε ο χρήστης. Αν ο χρήστης επιλέξει ως μέθοδο την 1, τότε θα εκτελεστεί η μέθοδος Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe με το αντίστοιχο αρχικό διάνυσμα. Αν ο χρήστης επιλέξει ως μέθοδο τη 2, τότε θα εκτελεστεί η μέθοδος Newton ασφαλούς περιοχής με το αντίστοιχο αρχικό διάνυσμα. Αν ο χρήστης επιλέξει ως μέθοδο την 3, τότε θα εκτελεστεί η μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe με το αντίστοιχο αρχικό διάνυσμα. Αν ο χρήστης επιλέξει ως μέθοδο την 4, τότε θα εκτελεστεί η μέθοδος steepest descent με ευθύγραμμη αναζήτηση με συνθήκες Wolfe με το αντίστοιχο αρχικό διάνυσμα. Επίσης, η συνάρτηση υπολογίζει το χρόνο σε δευτερόλεπτα που απαιτήθηκε για την υλοποίηση της μεθόδου. Φυσικά ο χρόνος αυτός διαφέρει και εξαρτάται από τον υπολογιστή στον οποίο τρέχει το πρόγραμμα.

Η `mean_squared_error(y_true, y_pred)` υπολογίζει το μέσο τετραγωνικό σφάλμα για την αξιολόγηση της ποιότητας πρόβλεψης του μοντέλου μας. Παίρνει ως είσοδο το διάνυσμα των πραγματικών τιμών  $Y$  και το διάνυσμα των τιμών που προβλέπει το μοντέλο μας  $\hat{Y}$ . Η συνάρτηση επιστρέφει το  $MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$ .

Η `plot_model_results(parameters)` δημιουργεί τη γραφική παράσταση παίρνοντας ως είσοδο τις παραμέτρους που προκύπτουν από τη συνάρτηση `optimizer()`. Πρώτα, υπολογίζει το μέσο τετραγωνικό σφάλμα του test set καλώντας τη συνάρτηση `mean_squared_error()`. Έπειτα, κατασκευάζει μια γραφική παράσταση που στον άξονα των x βρίσκονται οι μέρες από τις 14/10/20 και για τις επόμενες 35 ημέρες και στον άξονα των y βρίσκεται ο συσσωρευτικός αριθμός κρουσμάτων του covid-19 στην Ελλάδα. Με κόκκινες τελείες αναπαρίστανται τα δοθέντα δεδομένα που αποτέλεσαν το train set μας, ενώ με μπλε τελείες αναπαρίστανται τα δεδομένα για τα οποία υλοποιούμε την πρόβλεψη, δηλαδή αποτελούν το test set μας. Με μπλε γραμμή αναπαριστάται το μοντέλο μας για το σύνολο εκπαίδευσης, ενώ με πορτοκαλί γραμμή αναπαριστάται το προβλεπόμενο μοντέλο. Επιλέξαμε να κάνουμε μόνο μία γραφική παράσταση στην οποία θα περιλαμβάνεται και το train και το test set για να είναι πιο εύκολα διαχειρίσιμη η πληροφορία από το χρήστη.

### 3.2.4. Η κύρια συνάρτηση

Η εργασία μας έχει κατασκευαστεί ώστε να τρέχει μέσω τερματικού. Ο χρήστης θα πρέπει αρχικά να έχει το αρχείο της εργασίας μας μαζί με το αρχείο των δεδομένων στον ίδιο φάκελο. Στη συνέχεια, μπορεί να εκτελέσει τις μεθόδους που υλοποιήσαμε καταχωρώντας την κατάλληλη εντολή στο τερματικό του. Πιο συγκεκριμένα, ο χρήστης γράφει `optimization.py m a x b`, όπου το `a` αναφέρεται στον αριθμό της μεθόδου που επέλεξε και το `b` είναι το αρχικό διάνυσμα που επέλεξε. Σε περίπτωση που ο χρήστης καταχωρήσει λανθασμένα την εντολή ενημερώνεται ότι έχει δώσει λανθασμένη είσοδο και ενημερώνεται αναλυτικά για τις διαθέσιμες επιλογές. Η συνάρτηση `main()` δέχεται την καταχώρηση του χρήστη και καλεί τη συνάρτηση `optimizer()` για να ξεκινήσει η μέθοδος βελτιστοποίησης και έπειτα καλεί τη συνάρτηση `plot_model_results()` για να απεικονιστεί η γραφική αναπαράσταση του μοντέλου.



## Κεφάλαιο 4

### **Αποτελέσματα**

Στηριζόμενοι στο δοθέν αρχείο 'initial\_points.dat' υπάρχουν 5 αρχικές τιμές του διανύσματος

$\alpha = (a_0, a_1, a_2, a_3, a_4)^T$  των συντελεστών του μοντέλου για τις οποίες εφαρμόστηκαν οι διαδικασίες που περιγράφηκαν στο κεφάλαιο της υλοποίησης.

#### **4.1.Αποτελέσματα για την πρώτη περίπτωση**

$$\alpha = (5.025, -4.898, 0.119, 3.981, 7.818)^T$$

Αρχικά, εφαρμόσαμε τη μέθοδο Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Προέκυψε ότι η λύση βρέθηκε με ακρίβεια  $\varepsilon = 10^{-6}$  σε 2 επαναλήψεις με συνάρτηση σφάλματος ίση με 0.0391.

Στη συνέχεια εφαρμόσαμε τη μέθοδο Newton ασφαλούς περιοχής. Προέκυψε ότι η λύση βρέθηκε με ακρίβεια  $\varepsilon = 10^{-6}$  σε 17 επαναλήψεις. Επιλέχθηκαν συνολικά 1 Newton σημείο, 9 Cauchy σημεία και 7 Dogleg σημεία. Η συνάρτηση σφάλματος στην πρώτη επανάληψη ήταν ίση με 1827970.9285 και μειωνόταν αρκετά γρήγορα καταλήγοντας στην 17<sup>η</sup> επανάληψη να ναι ίση με 0.0391.

Ακολούθησε η μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Προέκυψε ότι η λύση βρέθηκε με ακρίβεια  $\varepsilon = 10^{-6}$  σε 11 επαναλήψεις. Η συνάρτηση σφάλματος στην πρώτη επανάληψη ήταν ίση με 426.4237 και μειώθηκε εξαιρετικά απότομα καταλήγοντας στην 11<sup>η</sup> επανάληψη να είναι ίση με 0.0391.

Τέλος, υλοποιήθηκε η μέθοδος steepest descent με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Παρατηρήσαμε ότι η μέθοδος ξεκίνησε με τη συνάρτηση σφάλματος να είναι ίση με 426.4237. Στις πρώτες επαναλήψεις υπήρξε μείωση της συνάρτησης σφάλματος, αλλά στη συνέχεια μειωνόταν απειροελάχιστα σε κάθε επανάληψη. Αυτό είχε ως αποτέλεσμα να εξαντληθούν οι 1000 επαναλήψεις που είχαμε θέσει ως περιορισμό και να καταλήξουμε στη χλυστή επανάληψη η συνάρτηση σφάλματος να ναι ίση με 16.2311. Επιλέξαμε να δοκιμάσουμε τι θα συμβεί σε περίπτωση που δεν υπάρχει το όριο των 1000 επαναλήψεων και διαπιστώσαμε ότι μετά από 134863 επαναλήψεις (το πρόγραμμα έτρεχε περίπου 20 ώρες) η συνάρτηση σφάλματος έγινε ίση με 0.172.

Συμπεραίνουμε δηλαδή ότι σε αυτή την περίπτωση το βέλτιστο μοντέλο ως προς το πλήθος των επαναλήψεων για την εύρεση της λύσης είναι με τη μέθοδο Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Οι παράμετροι του μοντέλου μετά τη βελτιστοποίηση είναι οι  $(2.19714047, 0.88569895, -0.80419248, 0.60753577, -0.09500749)^T$ .

#### **4.2.Αποτελέσματα για τη δεύτερη περίπτωση**

$$\alpha = (9.185, 0.944, -7.227, -7.014, -4.849)^T$$

Αρχικά, εφαρμόσαμε τη μέθοδο Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Προέκυψε ότι η λύση βρέθηκε με ακρίβεια  $\varepsilon = 10^{-6}$  σε 1 επανάληψη με συνάρτηση σφάλματος ίση με 0.0391.

Στη συνέχεια εφαρμόσαμε τη μέθοδο Newton ασφαλούς περιοχής. Προέκυψε ότι η λύση βρέθηκε με ακρίβεια  $\varepsilon = 10^{-6}$  σε 21 επαναλήψεις. Επιλέχθηκαν συνολικά 1 Newton σημείο, 10 Cauchy σημεία και 10 Dogleg σημεία. Η συνάρτηση σφάλματος στην πρώτη επανάληψη ήταν ίση με 1532847.7840 και μειωνόταν αρκετά γρήγορα καταλήγοντας στην 21<sup>η</sup> επανάληψη να ναι ίση με 0.0391.

Ακολούθησε η μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Προέκυψε ότι η λύση βρέθηκε με ακρίβεια  $\varepsilon = 10^{-6}$  σε 10 επαναλήψεις. Η συνάρτηση σφάλματος στην πρώτη επανάληψη ήταν ίση με 2833.2783 και μειώθηκε εξαιρετικά απότομα καταλήγοντας στη 10<sup>η</sup> επανάληψη να είναι ίση με 0.0391.

Τέλος, υλοποιήθηκε η μέθοδος steepest descent με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Παρατηρήσαμε ότι η μέθοδος ξεκίνησε με τη συνάρτηση σφάλματος να είναι ίση με 2833.2783 και μειωνόταν ελάχιστα σε κάθε επανάληψη. Αυτό είχε ως αποτέλεσμα να εξαντληθούν οι 1000 επαναλήψεις που είχαμε θέσει ως περιορισμό και να καταλήξουμε στη χλίοστη επανάληψη η συνάρτηση σφάλματος να ναι ίση με 57.752.

Συμπεραίνουμε δηλαδή ότι σε αυτή την περίπτωση το βέλτιστο μοντέλο ως προς το πλήθος των επαναλήψεων για την εύρεση της λύσης είναι με τη μέθοδο Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Οι παράμετροι του μοντέλου μετά τη βελτιστοποίηση είναι οι  $(2.19714047, 0.88569895, -0.80419248, 0.60753577, -0.09500749)^T$ .

#### 4.3.Αποτελέσματα για την τρίτη περίπτωση

$$\alpha = (6.814, -4.914, 6.285, -5.129, 8.585)^T$$

Αρχικά, εφαρμόσαμε τη μέθοδο Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Προέκυψε ότι η λύση βρέθηκε με ακρίβεια  $\varepsilon = 10^{-6}$  σε 2 επαναλήψεις με συνάρτηση σφάλματος ίση με 0.0391.

Στη συνέχεια εφαρμόσαμε τη μέθοδο Newton ασφαλούς περιοχής. Προέκυψε ότι η λύση βρέθηκε με ακρίβεια  $\varepsilon = 10^{-6}$  σε 26 επαναλήψεις. Επιλέχθηκαν συνολικά 1 Newton σημείο, 12 Cauchy σημεία και 13 Dogleg σημεία. Η συνάρτηση σφάλματος στην πρώτη επανάληψη ήταν ίση με 1168049.7183 και μειωνόταν αρκετά γρήγορα καταλήγοντας στην 26<sup>η</sup> επανάληψη να ναι ίση με 0.0391.

Ακολούθησε η μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Προέκυψε ότι η λύση βρέθηκε με ακρίβεια  $\varepsilon = 10^{-6}$  σε 13 επαναλήψεις. Η συνάρτηση σφάλματος στην πρώτη επανάληψη ήταν ίση με 1275.3895 και μειώθηκε εξαιρετικά απότομα καταλήγοντας στη 13<sup>η</sup> επανάληψη να είναι ίση με 0.0391.

Τέλος, υλοποιήθηκε η μέθοδος steepest descent με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Παρατηρήσαμε ότι η μέθοδος ξεκίνησε με τη συνάρτηση σφάλματος να είναι ίση με 1275.3895 και μειωνόταν ελάχιστα σε κάθε επανάληψη. Αυτό είχε ως αποτέλεσμα να εξαντληθούν οι 1000 επαναλήψεις που είχαμε θέσει ως περιορισμό και να καταλήξουμε στη χλίοστη επανάληψη η συνάρτηση σφάλματος να ναι ίση με 68.5231.

Συμπεραίνουμε δηλαδή ότι σε αυτή την περίπτωση το βέλτιστο μοντέλο ως προς το πλήθος των επαναλήψεων για την εύρεση της λύσης είναι με τη μέθοδο Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Οι παράμετροι του μοντέλου μετά τη βελτιστοποίηση είναι οι  $(2.19714047, 0.88569895, -0.80419248, 0.60753577, -0.09500749)^T$ .

#### 4.4.Αποτελέσματα για την τέταρτη περίπτωση

$$\alpha = (-2.966, 6.616, 1.705, 0.994, 8.343)^T$$

Αρχικά, εφαρμόσαμε τη μέθοδο Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Προέκυψε ότι η λύση βρέθηκε με ακρίβεια  $\varepsilon = 10^{-6}$  σε 2 επαναλήψεις με συνάρτηση σφάλματος ίση με 0.0391.

Στη συνέχεια εφαρμόσαμε τη μέθοδο Newton ασφαλούς περιοχής. Προέκυψε ότι η λύση βρέθηκε με ακρίβεια  $\varepsilon = 10^{-6}$  σε 19 επαναλήψεις. Επιλέχθηκαν συνολικά 1 Newton σημείο, 10 Cauchy σημεία και 8 Dogleg σημεία. Η συνάρτηση σφάλματος στην πρώτη επανάληψη ήταν ίση με 1865479.5907 και μειωνόταν αρκετά γρήγορα καταλήγοντας στη 19<sup>η</sup> επανάληψη να ναι ίση με 0.0391.

Ακολούθησε η μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Προέκυψε ότι η λύση βρέθηκε με ακρίβεια  $\varepsilon = 10^{-6}$  σε 12 επαναλήψεις. Η συνάρτηση σφάλματος στην πρώτη επανάληψη ήταν ίση με 178.4908 και μειώθηκε καταλήγοντας στη 12<sup>η</sup> επανάληψη να είναι ίση με 0.0391.

Τέλος, υλοποιήθηκε η μέθοδος steepest descent με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Παρατηρήσαμε ότι η μέθοδος ξεκίνησε με τη συνάρτηση σφάλματος να είναι ίση με

178.4908. Στη δεύτερη επανάληψη υπήρξε μεγάλη μείωση της συνάρτησης σφάλματος, αλλά στη συνέχεια η συνάρτηση μειωνόταν ελάχιστα σε κάθε επανάληψη. Αυτό είχε ως αποτέλεσμα να εξαντληθούν οι 1000 επαναλήψεις που είχαμε θέσει ως περιορισμό και να καταλήξουμε στη χιλιοστή επανάληψη η συνάρτηση σφάλματος να ναι ίση με 31.7139. Συμπεραίνουμε δηλαδή ότι σε αυτή την περίπτωση το βέλτιστο μοντέλο ως προς το πλήθος των επαναλήψεων για την εύρεση της λύσης είναι με τη μέθοδο Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Οι παράμετροι του μοντέλου μετά τη βελτιστοποίηση είναι οι  $(2.19714047, 0.88569895, -0.80419248, 0.60753577, -0.09500749)^T$ .

#### 4.5.Αποτελέσματα για την πέμπτη περίπτωση

$$\alpha = (-7.401, 1.376, -0.612, -9.762, -3.257)^T$$

Αρχικά, εφαρμόσαμε τη μέθοδο Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Προέκυψε ότι η λύση βρέθηκε με ακρίβεια  $\varepsilon = 10^{-6}$  σε 1 επανάληψη με συνάρτηση σφάλματος ίση με 0.0391.

Στη συνέχεια εφαρμόσαμε τη μέθοδο Newton ασφαλούς περιοχής. Προέκυψε ότι η λύση βρέθηκε με ακρίβεια  $\varepsilon = 10^{-6}$  σε 21 επαναλήψεις. Επιλέχθηκαν συνολικά 1 Newton σημείο, 7 Cauchy σημεία και 13 Dogleg σημεία. Η συνάρτηση σφάλματος στην πρώτη επανάληψη ήταν ίση με 1112286.2293 και μειωνόταν αρκετά γρήγορα καταλήγοντας στην 21<sup>η</sup> επανάληψη να ναι ίση με 0.0391.

Ακολούθησε η μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Προέκυψε ότι η λύση βρέθηκε με ακρίβεια  $\varepsilon = 10^{-6}$  σε 12 επαναλήψεις. Η συνάρτηση σφάλματος στην πρώτη επανάληψη ήταν ίση με 6723.6606 και μειώθηκε εξαιρετικά απότομα καταλήγοντας στη 12<sup>η</sup> επανάληψη να είναι ίση με 0.0391.

Τέλος, υλοποιήθηκε η μέθοδος steepest descent με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Παρατηρήσαμε ότι η μέθοδος ξεκίνησε με τη συνάρτηση σφάλματος να είναι ίση με 6723.6606 και μειωνόταν ελάχιστα σε κάθε επανάληψη. Αυτό είχε ως αποτέλεσμα να εξαντληθούν οι 1000 επαναλήψεις που είχαμε θέσει ως περιορισμό και να καταλήξουμε στη χιλιοστή επανάληψη η συνάρτηση σφάλματος να ναι ίση με 53.8412.

Συμπεραίνουμε δηλαδή ότι σε αυτή την περίπτωση το βέλτιστο μοντέλο ως προς το πλήθος των επαναλήψεων για την εύρεση της λύσης είναι με τη μέθοδο Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Οι παράμετροι του μοντέλου μετά τη βελτιστοποίηση είναι οι  $(2.19714047, 0.88569895, -0.80419248, 0.60753577, -0.09500749)^T$ .

#### 4.6.Σύνοψη αποτελεσμάτων

Στον επόμενο πίνακα συνοψίζεται το πλήθος των επαναλήψεων που απαιτήθηκαν σε κάθε μέθοδο με βάση το αρχικό διάνυσμα συντελεστών που επιλέχθηκε.

$\alpha$	Η μέθοδος Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe	Η μέθοδος Newton ασφαλούς περιοχής	Η μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe	Η μέθοδος steepest descent με ευθύγραμμη αναζήτηση με συνθήκες Wolfe
1 <sup>ο</sup> διάνυσμα	2	17	11	1000
2 <sup>ο</sup> διάνυσμα	1	21	10	1000
3 <sup>ο</sup> διάνυσμα	2	26	13	1000
4 <sup>ο</sup> διάνυσμα	2	19	12	1000
5 <sup>ο</sup> διάνυσμα	1	21	12	1000

Παρατηρήσαμε ότι σε κάθε μία από τις 5 περιπτώσεις η μέθοδος Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe αποδείχτηκε η μέθοδος που απαιτούσε τις λιγότερες επαναλήψεις για την εύρεση της λύσης, ενώ και οι 4 μέθοδοι κατάφεραν να μειώσουν την αρχική συνάρτηση σφάλματος. Η δεύτερη καλύτερη μέθοδος ήταν η μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe, η οποία παρουσίαζε πολύ απότομη μείωση στη συνάρτηση σφάλματος. Η τρίτη καλύτερη μέθοδος ήταν η μέθοδος Newton ασφαλούς περιοχής, η οποία τερμάτιζε μόλις εντόπιζε το Newton point. Η πιο αργή μέθοδος ήταν η μέθοδος steepest descent με ευθύγραμμη αναζήτηση με συνθήκες Wolfe, η οποία στις πρώτες επαναλήψεις εμφάνιζε μια αρκετά σημαντική μείωση της συνάρτησης σφάλματος, αλλά στις επόμενες επαναλήψεις η μείωση ήταν απειροελάχιστη, οπότε τερμάτιζε λόγω εξαντλήσεως των επιτρεπτών επαναλήψεων.

Ακόμη, οι πρώτες τρεις μέθοδοι μας οδήγησαν στο ίδιο διάνυσμα παραμέτρων του μοντέλου μετά τη βελτιστοποίηση και στις 5 περιπτώσεις, όπως περιμέναμε αφού εντοπίσανε τον ελαχιστοποιητή. Αντίθετα, η μέθοδος steepest descent με ευθύγραμμη αναζήτηση με συνθήκες Wolfe μας οδήγησε σε διαφορετικά διανύσματα. Πιο συγκεκριμένα, στην πρώτη περίπτωση το διάνυσμα ήταν ίσο με  $(4.83034144, -4.29599169, 0.19834694, 1.84034354, -0.46910162)^T$ . Στη δεύτερη περίπτωση το διάνυσμα ήταν ίσο με  $(4.76872515, 1.33335995, -2.86538735, 0.921241941, -0.00296097448)^T$ . Στην τρίτη περίπτωση το διάνυσμα ήταν ίσο με  $(5.51761326, -4.69342403, 7.37692598, -4.74018405, 0.97392848)^T$ . Στην τέταρτη περίπτωση το διάνυσμα ήταν ίσο με  $(-1.28869402, 6.15222827, 0.21193923, -2.09385092, 0.55530082)^T$ . Στη πέμπτη περίπτωση το διάνυσμα ήταν ίσο με  $(-1.96533506, 5.00820868, 3.68641896, -4.15489449, 0.90968711)^T$ . Τα παραπάνω διανύσματα ήταν διαφορετικά από το διάνυσμα που προέκυψε από τις υπόλοιπες 3 μεθόδους, καθώς η μέθοδος τερματίστηκε λόγω της εξάντλησης των 1000 επαναλήψεων και όχι λόγω εύρεσης του ελαχιστοποιητή.

## Κεφάλαιο 5

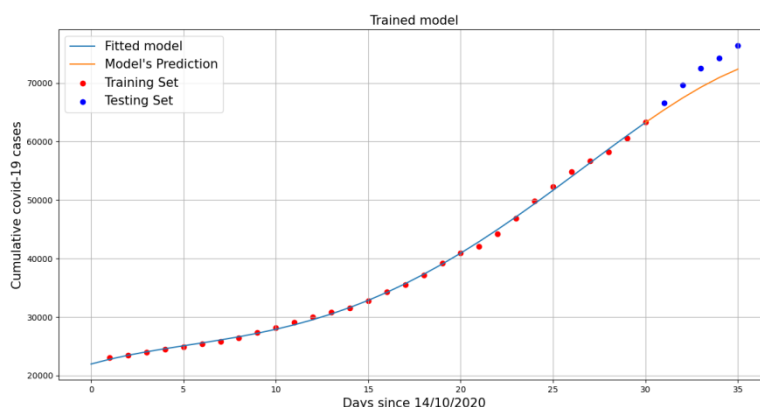
### Προβλέψεις

Στο κεφάλαιο αυτό θα χρησιμοποιηθεί το βέλτιστο μοντέλο σε κάθε μία από τις 5 περιπτώσεις προκειμένου να προβλεφθούν οι τιμές των κρουσμάτων για 5 ημέρες μετά από το πέρας των δεδομένων, δηλαδή από 13 έως 17 Νοεμβρίου 2020. Οι τιμές αυτές θα συγκριθούν με τα πραγματικά δεδομένα με τα κρούσματα, όπως προκύπτουν από την ιστοσελίδα <https://covid19.who.int/table> και θα εξαχθούν συμπεράσματα ως προς την ποιότητα πρόβλεψης του μοντέλου, στηριζόμενοι στη μετρική του μέσου τετραγωνικού σφάλματος, που ορίζεται ως  $MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$ , όπου  $Y$  το διάνυσμα των πραγματικών τιμών και  $\hat{Y}$  το διάνυσμα των τιμών που προβλέπει το μοντέλο μας.

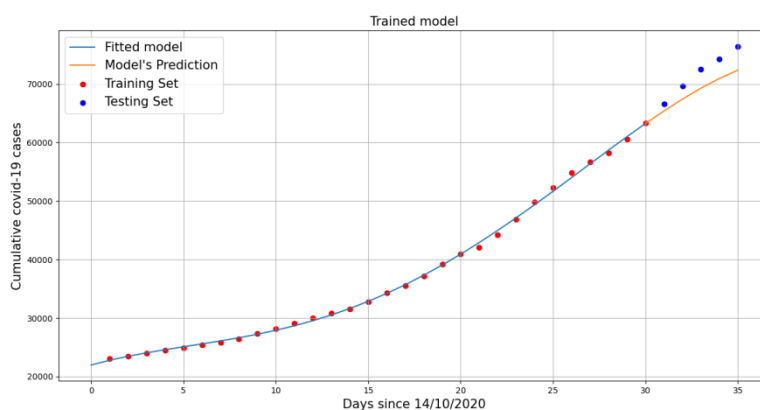
#### 5.1. Προβλέψεις για την πρώτη περίπτωση

Υπολογίζοντας το μέσο τετραγωνικό σφάλμα για το test set και στις 4 μεθόδους διαπιστώσαμε ότι το μέσο τετραγωνικό σφάλμα ήταν μικρότερο στη μέθοδο Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe, στη μέθοδο Newton ασφαλούς περιοχής, αλλά και στη μέθοδο BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Τα γραφήματα που ακολουθούν αναπαριστούν το train και το test set στις 3 μεθόδους.

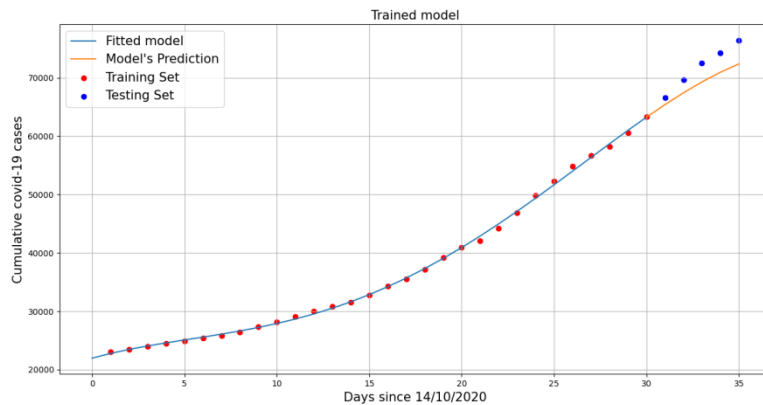
Μέθοδος Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe:



Μέθοδος Newton ασφαλούς περιοχής:



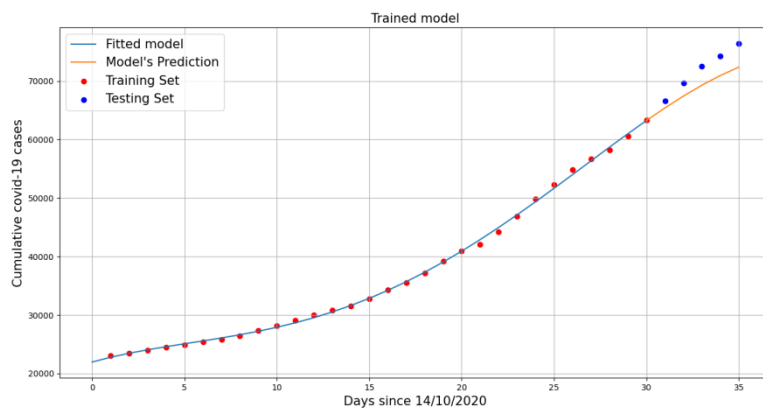
Μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe:



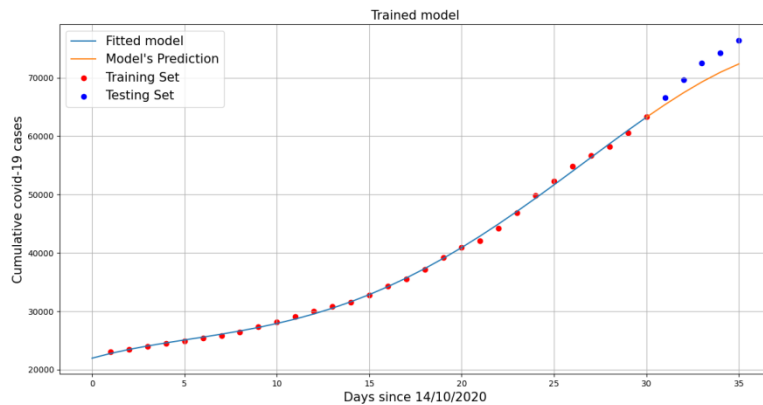
## 5.2. Προβλέψεις για τη δεύτερη περίπτωση

Υπολογίζοντας το μέσο τετραγωνικό σφάλμα για το test set και στις 4 μεθόδους διαπιστώσαμε ότι το μέσο τετραγωνικό σφάλμα ήταν μικρότερο στη μέθοδο Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe, στη μέθοδο Newton ασφαλούς περιοχής, αλλά και στη μέθοδο BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Τα γραφήματα που ακολουθούν αναπαριστούν το train και το test set στις 3 μεθόδους.

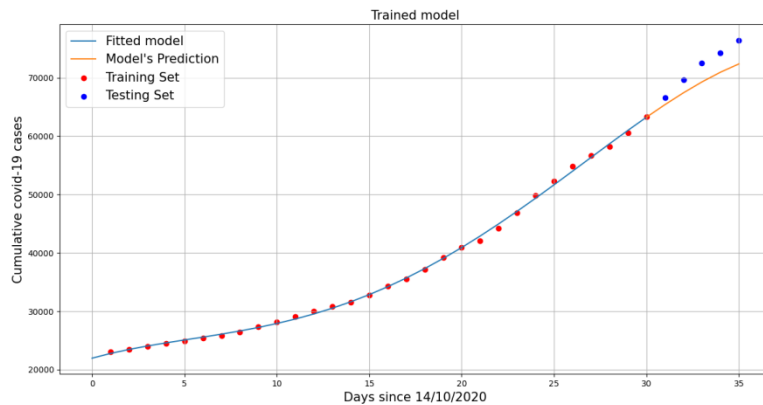
Μέθοδος Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe:



Μέθοδος Newton ασφαλούς περιοχής:



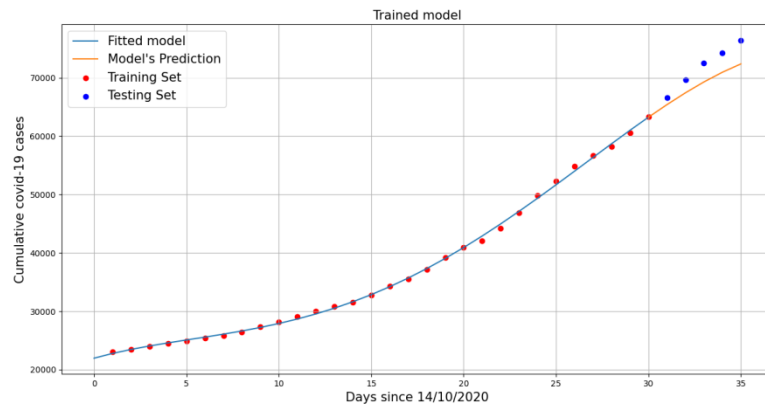
Μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe:



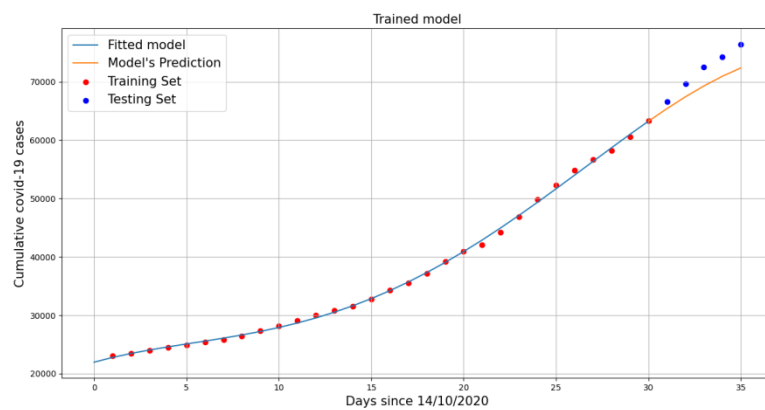
### 5.3. Προβλέψεις για την τρίτη περίπτωση

Υπολογίζοντας το μέσο τετραγωνικό σφάλμα για το test set και στις 4 μεθόδους διαπιστώσαμε ότι το μέσο τετραγωνικό σφάλμα ήταν μικρότερο στη μέθοδο Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe, στη μέθοδο Newton ασφαλούς περιοχής, αλλά και στη μέθοδο BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Τα γραφήματα που ακολουθούν αναπαριστούν το train και το test set στις 3 μεθόδους.

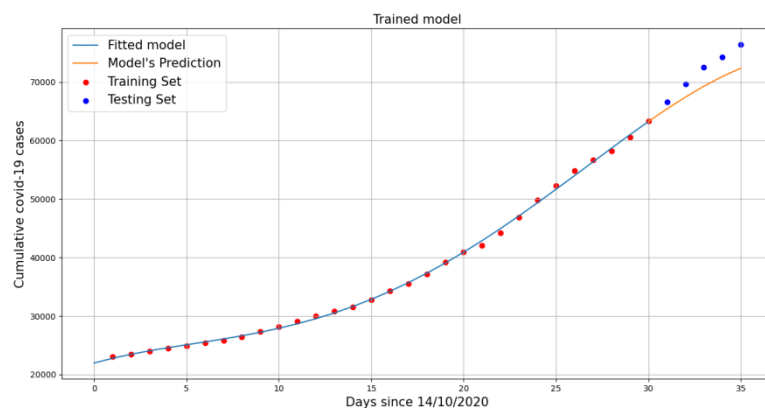
Μέθοδος Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe:



Μέθοδος Newton ασφαλούς περιοχής:



Μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe:



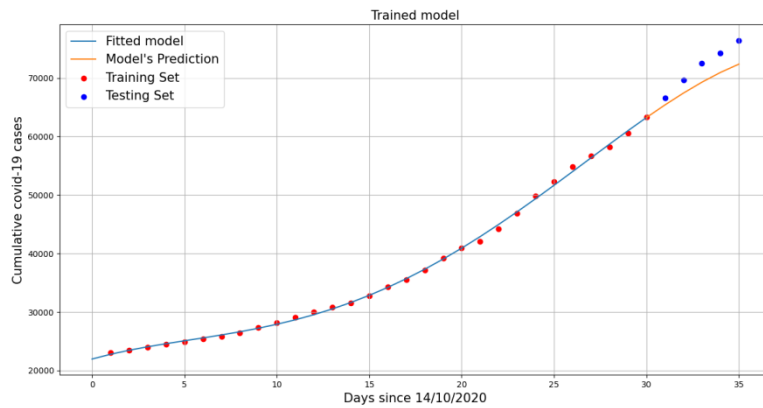
#### 5.4. Προβλέψεις για την τέταρτη περίπτωση

Υπολογίζοντας το μέσο τετραγωνικό σφάλμα για το test set και στις 4 μεθόδους διαπιστώσαμε ότι το μέσο τετραγωνικό σφάλμα ήταν μικρότερο στη μέθοδο Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe, στη μέθοδο Newton ασφαλούς περιοχής, αλλά

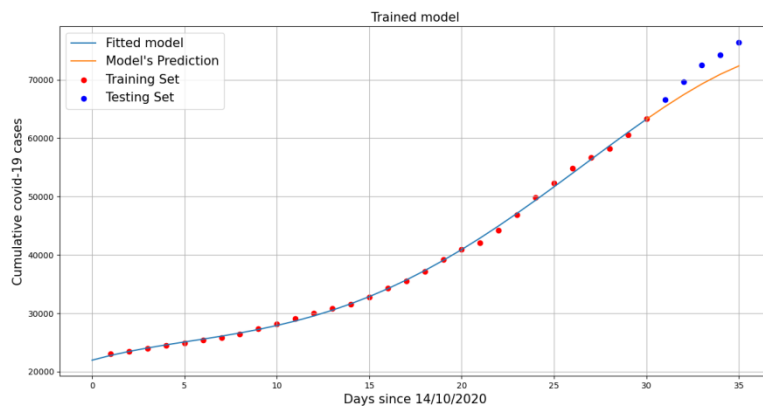


και στη μέθοδο BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Τα γραφήματα που ακολουθούν αναπαριστούν το train και το test set στις 3 μεθόδους.

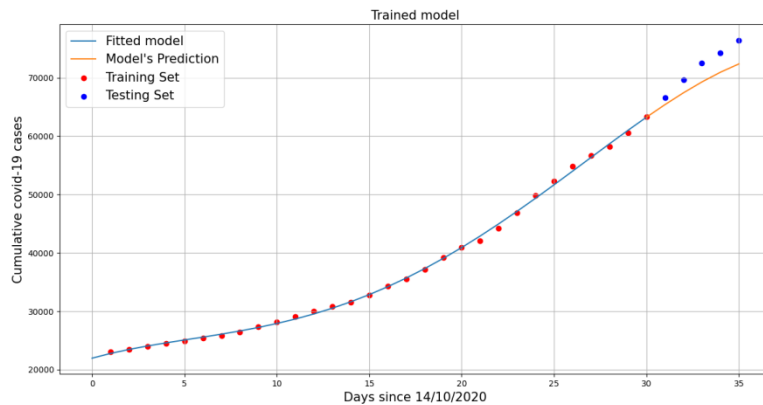
Μέθοδος Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe:



Μέθοδος Newton ασφαλούς περιοχής:



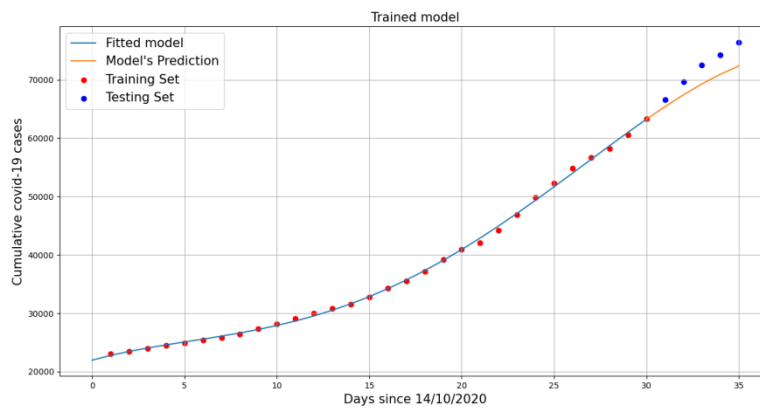
Μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe:



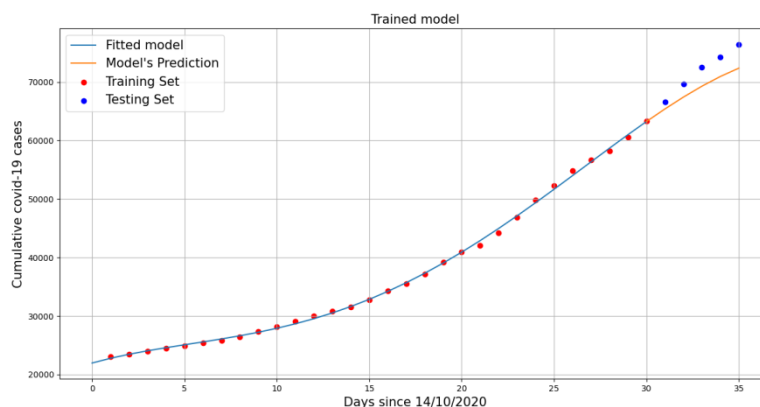
### 5.5. Προβλέψεις για την πέμπτη περίπτωση

Υπολογίζοντας το μέσο τετραγωνικό σφάλμα για το test set και στις 4 μεθόδους διαπιστώσαμε ότι το μέσο τετραγωνικό σφάλμα ήταν μικρότερο στη μέθοδο Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe, στη μέθοδο Newton ασφαλούς περιοχής, αλλά και στη μέθοδο BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Τα γραφήματα που ακολουθούν αναπαριστούν το train και το test set στις 3 μεθόδους.

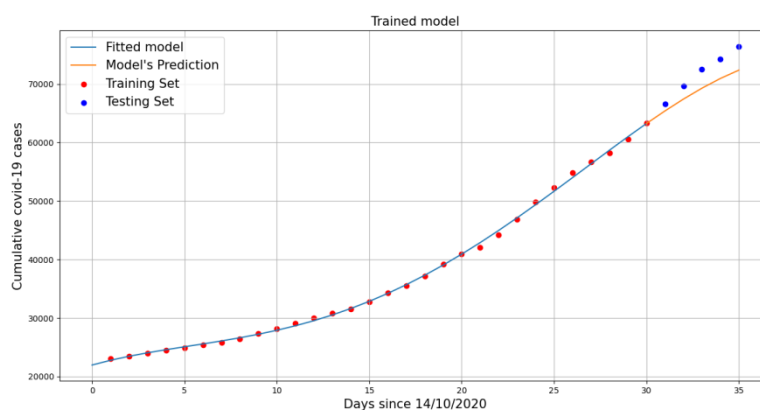
Μέθοδος Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe:



Μέθοδος Newton ασφαλούς περιοχής:

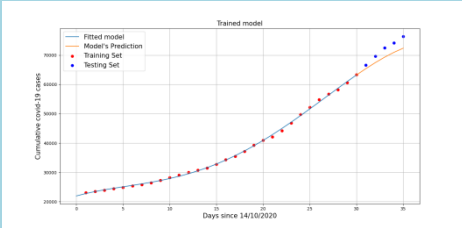
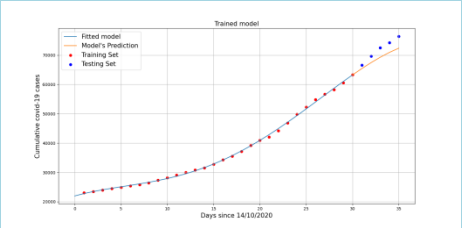
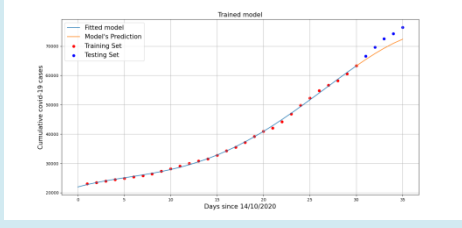
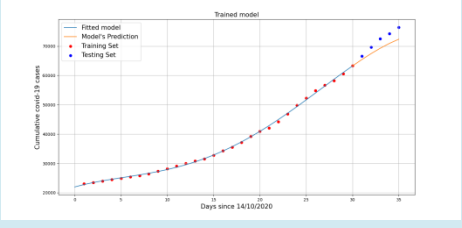
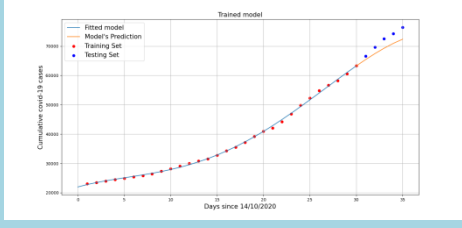
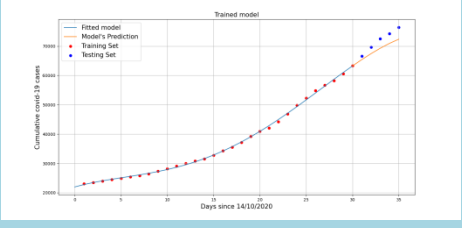
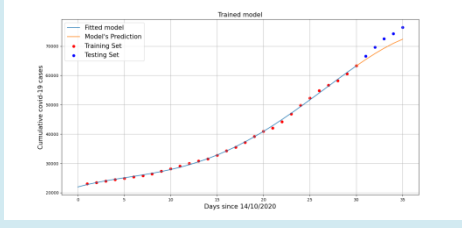
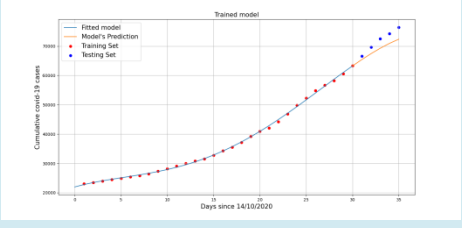
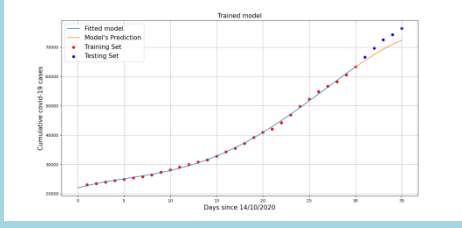
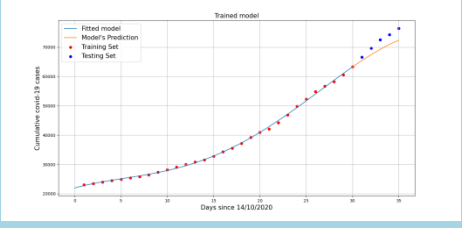


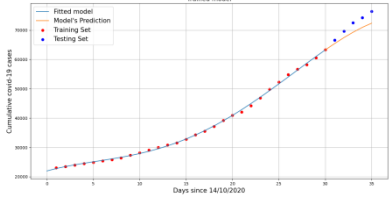
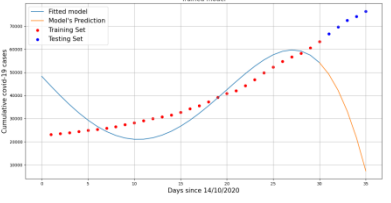
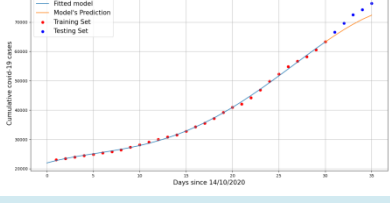
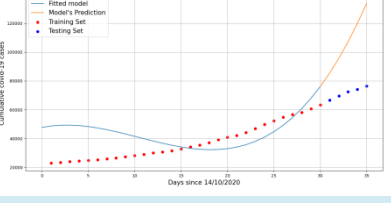
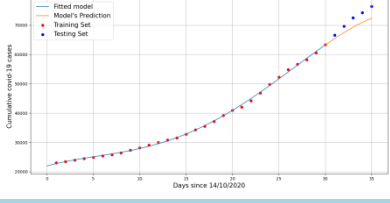
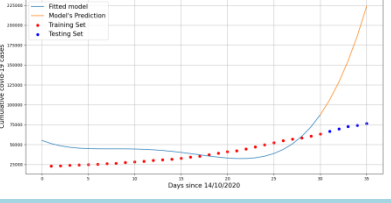
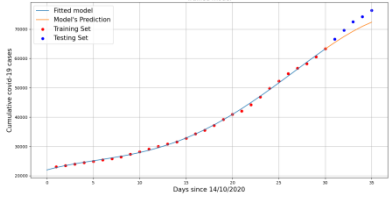
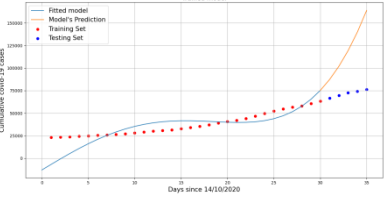
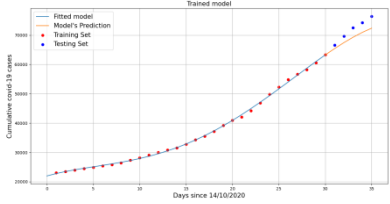
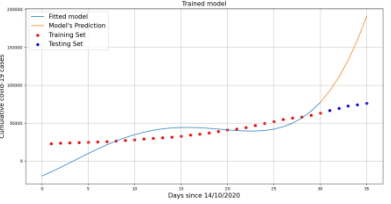
Μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe:



## 5.6.Σύνοψη των προβλέψεων

Στους επόμενους 2 πίνακες ενσωματώσαμε τις γραφικές παραστάσεις των μοντέλων που προέκυψαν σε κάθε μέθοδο, ξεκινώντας με τα 5 διαφορετικά αρχικά διανύσματα των συντελεστών του μοντέλου.

$\alpha$	Η μέθοδος Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe	Η μέθοδος Newton ασφαλούς περιοχής
1 <sup>η</sup>		
2 <sup>η</sup>		
3 <sup>η</sup>		
4 <sup>η</sup>		
5 <sup>η</sup>		

α	Η μέθοδος BFGS με ερυθράγραμμα αναζήτηση με συνθήκες Wolfe	Η μέθοδος steepest descent με ερυθράγραμμα αναζήτηση με συνθήκες Wolfe
1 <sup>η</sup>		
2 <sup>η</sup>		
3 <sup>η</sup>		
4 <sup>η</sup>		
5 <sup>η</sup>		

Στον επόμενο πίνακα καταγράψαμε το μέσο τετραγωνικό σφάλμα στο test set σε κάθε περίπτωση. Το μέσο τετραγωνικό σφάλμα το υπολογίσαμε χρησιμοποιώντας τα scaled data, καθώς αρχικά το είχαμε υπολογίσει πάνω στα δοθέντα δεδομένα και δημιουργούνταν πρόβλημα υπερχειλίσης. Πιο συγκεκριμένα, το πρόβλημα εντοπίζονταν στην τέταρτη μέθοδο που προέκυπε αρνητικό μέσο τετραγωνικό σφάλμα, κάτι το οποίο ήταν αδύνατο.

MSE (at test set)	Η μέθοδος Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe	Η μέθοδος Newton ασφαλούς περιοχής	Η μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe	Η μέθοδος steepest descent με ευθύγραμμη αναζήτηση με συνθήκες Wolfe
1 <sup>η</sup> περίπτωση	0.0879	0.0879	0.0879	20.2680
2 <sup>η</sup> περίπτωση	0.0879	0.0879	0.0879	15.2505
3 <sup>η</sup> περίπτωση	0.0879	0.0879	0.0879	92.3911
4 <sup>η</sup> περίπτωση	0.0879	0.0879	0.0879	31.2603
5 <sup>η</sup> περίπτωση	0.0879	0.0879	0.0879	52.6292

Στον πίνακα παρατηρούμε ότι το μέσο τετραγωνικό σφάλμα είναι ίσο στις 3 πρώτες μεθόδους για τα 5 δοθέντα αρχικά διανύσματα των συντελεστών του μοντέλου. Στην τέταρτη μέθοδο το μέσο τετραγωνικό σφάλμα διαφέρει και μάλιστα είναι αρκετά μεγαλύτερο.

## Κεφάλαιο 6

### Παράρτημα

**Σύγκριση των 4 μεθόδων βελτιστοποίησης ως προς τον χρόνο σε δευτερόλεπτα που απαιτήσαν.**

Τέλος, αν θελήσουμε να συγκρίνουμε τις 4 μεθόδους ως προς το χρόνο που απαιτήθηκε για να ολοκληρωθούν σε κάθε μία από τις 5 περιπτώσεις του αρχικού διανύσματος θα διαπιστώσουμε ότι η μέθοδος Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe είναι τάχιστα. Την ακολουθούν η μέθοδος Newton ασφαλούς περιοχής που είναι επίσης γρήγορη και η μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe. Μακράν η πιο αργή μέθοδος είναι η μέθοδος steepest descent με ευθύγραμμη αναζήτηση με συνθήκες Wolfe, κάτι το οποίο είναι αναμενόμενο μιας και πραγματοποιεί τις περισσότερες επαναλήψεις.

Χρόνος σε δευτερόλεπτα	Η μέθοδος Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe	Η μέθοδος Newton ασφαλούς περιοχής	Η μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe	Η μέθοδος steepest descent με ευθύγραμμη αναζήτηση με συνθήκες Wolfe
1 <sup>η</sup> περίπτωση	0.26	2.08	7.19	1127.49
2 <sup>η</sup> περίπτωση	0.14	2.61	6.84	1206.02
3 <sup>η</sup> περίπτωση	0.26	3.17	19.07	1211.13
4 <sup>η</sup> περίπτωση	0.25	2.35	7.25	578.68
5 <sup>η</sup> περίπτωση	0.14	2.59	7.21	525.04