



Hochschule Neubrandenburg  
University of Applied Sciences

Hochschule Neubrandenburg  
Master Geoinformatik und Geodäsie

# Technische Dokumentation

Rekonstruktion ionosphärischer Schlüsselparameter  
der F2 - Schicht mittels Kriging



DLR Deutsches Zentrum  
für Luft- und Raumfahrt

vorgelegt von: *Torben Boje*

geboren am: *20.07.1990*

# Inhaltsverzeichnis

<b>1 EINLEITUNG .....</b>	<b>3</b>
<b>2 PROGRAMMSTRUKTUR.....</b>	<b>3</b>
2.1 KLASSEN .....	3
2.1.1 Reconstruction.py.....	3
2.1.2 OrdinaryKriging.py.....	9
2.1.3 MeasurementStations.py.....	16
2.1.4 KrigeValidation.py .....	20
2.1.5 VariogramInR.py .....	23

# 1 Einleitung

## 2 Programmstruktur

### 2.1 Klassen

In diesem Kapitel werden die Klassen des Kriging Moduls vorgestellt. Ausgenommen sind externe Module, wie beispielsweise der NmF2/hmF2 Reader oder der Reader für die F10.7 Werte. Des Weiteren werden die Methoden und globalen Variablen erleutert.

#### 2.1.1 Reconstruction.py

Die Klasse Reconstruction.py verwaltet das komplette Kriging Modul. Dazu gehört das Auslesen der NmF2- und hmF2-Parameter, das berechnen der Modellwerte, das Auslesen der F10.7 Werte und die Validierung der Parameter aus der Konfigurationsdatei. Weitere Funktionen sind im folgenden erläutert.

#### Globale Variablen

self.Layer	„NmF2“, „hmF2“ oder eine Kombination aus den beiden Layern. Wert ist aus der Konfigurationsdatei.
self.DoValidation	Boolscher Wert, der angibt, ob die Validierung mit anderen Schätzverfahren durchgeführt werden soll (True) oder nicht (False). Wert ist aus der Konfigurationsdatei.
self.Resolution	Auflösung, mit der die Kriging Schätzung global durchgeführt werden soll. Der Wert gilt sowohl für Latitude und Longitude. Wert ist aus der Konfigurationsdatei.
self.PathToCompFileNmF2	Pfad zu der Datei, die Werte aus anderen Schätzverfahren für NmF2 beinhaltet. Wird nur benötigt, wenn self.Resolution True ist. Wert ist aus der Konfigurationsdatei.
self.PathToCompFilehmF2	Pfad zu der Datei, die Werte aus anderen Schätzverfahren für hmF2 beinhaltet. Wird nur

	benötigt, wenn self.Resolution True ist. Wert ist aus der Konfigurationsdatei.
self.RefStations	Liste der IDs der Referenzstationen. Diese Stationen werden bei der Variogrammschätzung berücksichtigt, aber nicht bei der Kriging Schätzung. Angabe erfolgt in Arrayform.
self.Outputfolder	Pfad in den alle Ergebnisdateien ausgegeben werden. Wert ist aus der Konfigurationsdatei.
self.LogFileName	Name der Logdatei.
self.Time	Uhrzeit im Format HH:MM. Wert ist aus der Konfigurationsdatei.
self.TimeIntervall	Zeitintervall (ZI), in dem bei den Messstationen nach Werten gesucht werden soll in Minuten. Für die Angabe gilt: $(ZI/2 \leq \text{Zeit} < (ZI/2))$ . Wert ist aus der Konfigurationsdatei.
self.Startdoy	Start Tag des Jahres. Wert ist aus der Konfigurationsdatei.
self.Enddoy	End Tag des Jahres. Wert ist aus der Konfigurationsdatei.
self.PathSpidr_hmF2	Pfad zu den hmF2-Daten des Providers SPIDR. Wert ist aus der Konfigurationsdatei.
self.PathDataAll_hmF2	Pfad zu dem <i>data_all</i> Verzeichnis, in dem die hmF2-Daten von allen Providern von <i>reader_ionodata.py</i> gespeichert werden. Wert ist aus der Konfigurationsdatei.
self.PathDidb_hmF2	Pfad zu den hmF2-Daten des Providers DIDB. Wert ist aus der Konfigurationsdatei.
self.PathNgdc_hmF2	Pfad zu den hmF2-Daten des Providers NGDC. Wert ist aus der Konfigurationsdatei.
self.PathSpidr_NmF2	Pfad zu den NmF2-Daten des Providers SPIDR. Wert ist aus der Konfigurationsdatei.
self.PathDataAll_NmF2	Pfad zu dem <i>data_all</i> Verzeichnis, in dem die NmF2-Daten von allen Providern von <i>reader_ionodata.py</i>

	gespeichert werden. Wert ist aus der Konfigurationsdatei.
self.PathDidb_NmF2	Pfad zu den NmF2-Daten des Providers DIDB. Wert ist aus der Konfigurationsdatei.
self.PathNgdc_NmF2	Pfad zu den NmF2-Daten des Providers NGDC. Wert ist aus der Konfigurationsdatei.
self.LagDistance	Lag Distanz aus der Konfigurationsdatei
self.PathF107	Pfad zu den F10.7 Dateien. Angegeben in der Konfigurationsdatei.
self.VarModel	Variogrammmodell. Angegeben in der Konfigurationsdatei. „sph“ für sphärisch, „exp“ für exponentielle, „gau“ für das Gaus'sche und „pow“ für das Power Modell. Soll die Kriging Schätzung für mehrere Variogrammmodelle durchgeführt werden, kann eine Kombination aus den aufgelisteten angegeben werden.
self.BackModel	Hintergrundmodell, angegeben in Kleinbuchstaben („nequick“, „npdm“ und „nphm“). Mehrere Angaben sind möglich.
self.Hour	Stunden des Tages. Bsp.: 9,5 für 09:30 Uhr
self.AltList	Liste mit Höhen im Intervall von 0km bis 700km in 1km Schritten. benötigt für die NeQuick Wertebestimmung.
self.F107	Liste für die F10.7 Werte.
self.Year	Jahr, aus dem die Messwerte stammen.
self.Month	Monat, aus dem die Messwerte stammen.
self.AllStations	MeasurementStation Objekt, in dem alle Messstationen verwaltet werden. Doppelt auftretende Messstationen werden nicht herausgefiltert.
self.UniqueStations	MeasurementStation Objekt, in dem alle Messstationen verwaltet werden. Doppelt auftretende Messstationen werden nicht mit aufgenommen.

## Methoden

<code>__init__()</code>	Konstruktor. Liest Werte aus der Konfigurationsdatei und überprüft diese.
<code>startAllParameter()</code>	Diese Methode wird aufgerufen, wenn bei <i>self.Layer</i> beide Layer angegeben werden. Die Methode startet nacheinander die Berechnung für NmF2 und hmF2.
<code>runReconstruction()</code>	Methode, die die komplette Verwaltung des Kriging Moduls übernimmt. Aus dieser Methode werden alle anderen benötigten Methoden aufgerufen. Diese Methode wird gestartet, nachdem der Konstruktor aufgerufen wird.
<code>readData()</code>	Diese Methode liest die NmF2- und hmF2-Werte aus und speichert sie in einem <i>MeasurementStation</i> -Objekt.
<code>doOK(model, variogrammodel)</code>	Methode, die die Ordinary Kriging Berechnung in der Klasse <i>OrdinaryKriging.py</i> initialisiert. <u>Übergabeparameter:</u> model: String, Hintergrundmodell variogrammodel: String, Variogrammmodell (keine Doppelnennungen)
<code>plotValueWithRespectToCoords (coordsUnique, coords_read, coordtyp)</code>	Diese Funktion stellt die ausgelesenen Messwerte in Abhängigkeit von der Latitude und Longitude dar. <u>Übergabeparameter:</u> coordsUnique: Liste der Latituden oder Longituden (einzigartig) coords_read: Liste der Latituden oder Longituden (nicht einzigartig) coordtyp: String, Latitude oder Longitude
<code>plotHistogram()</code>	Erstellt ein Histogramm der ausgelesenen Messwerte

plotNeQuickOneStation()	Stellt die Messwerte für das NeQuick Modell für einen Tag und für eine Station im Höhenintervall von 0km bis 600km dar.
plotNPDMorNPHMOneStation()	Stellt die Messwerte für das NPDM oder NPHM Modell für einen Tag und für eine Station dar. Welches Modell hängt dabei vom angegebenen Layer ab.
plotNeQuickDiffMeasurement()	Diese Funktion stellt die relativen Differenzen zwischen den Messwerten und den NeQuick Werten dar. <u>Return:</u> Liste mit den Residuen
plotDiff(coordsUnique, diff, filename, xlab, ylab)	Hilfsfunktion, die von plotNeQuickDiff Measurement() und plotNPDMorNPHMDiff Measurement() aufgerufen wird um die relativen Differenzen in einem Diagramm darzustellen. <u>Übergabeparameter:</u> coordsUnique: Liste der Latituden oder Longituden (einzigartig) diff: Liste der relativen Differenzen filename: Dateiname für die Darstellung xlab: Name der X-Achse ylab: Name der Y-Achse
plotStations()	Diese Funktion stellt alle Messstationen auf einer Weltkarte dar.
plotNPDMorNPHMDiffMeasurement()	Diese Funktion stellt die relativen Differenzen zwischen den Messwerten und den NPDM oder NPHM Werten dar. Welches Modell verwendet wird, ist abhängig vom Layer. <u>Return:</u> Liste mit den Residuen
getf107Values()	Diese Methode liest die F10.7-Werte aus den Dateien.

<p>getDayOfMonthandMonth(doy)</p>	<p>Hilfsfunktion, die aus dem Tag des Jahres den Monat und den Tag des Monats bestimmt.</p> <p><u>Übergabeparameter:</u> doy: Tag des Jahres</p> <p><u>Return:</u> Tag des Monats, Monat</p>
<p>getDayOfYear(dom, month, year)</p>	<p>Hilfsfunktion, die aus dem Tag des Monats, dem Monat und dem Jahr den Tag des Jahres berechnet.</p> <p><u>Übergabeparameter:</u> dom: Tag des Monats month: Monat year: Jahr</p> <p><u>Return:</u> Tag des Jahres</p>
<p>convertMinInHour(min)</p>	<p>Hilfsfunktion, die Minuten in Stunden umrechnet.</p> <p><u>Übergabeparameter:</u> min: Minuten</p> <p><u>Return:</u> Stunden</p>
<p>writelnLog(message)</p>	<p>Diese Funktion schreibt Informationen in die Log-Datei.</p> <p><u>Übergabeparameter:</u> message: String, Nachricht, die in die Log-Datei geschrieben werden soll.</p>
<p>writelnListofResults(filename)</p>	<p>Diese Funktion schreibt Dateinamen in die „ListOfResults.dat“, die alle Dateinamen von allen generierten Dateien enthält.</p> <p><u>Übergabeparameter:</u> filename: Dateiname</p>

### 2.1.2 OrdinaryKriging.py

Diese Klasse führt die Kriging Schätzung mit der Kriging Methode Ordinary Kriging durch. Des Weiteren werden in dieser Klasse die Variogramme geschätzt und Ausgabefunktionen zur Verfügung gestellt.

#### Globale Variablen

self.Stations	<i>MeasurementStation</i> -Objekt
self.Lats	Liste aller Latituden der Messstationen
self.Lons	Liste aller Longituden der Messstationen
self.DiffModelValues	Liste der Residuen
self.MeasurementValues	Liste aller Messwerte
self.EarthRadius	Erdradius in Kilometer (6378km)
self.Nugget	Wert des Variogrammparameters Nugget
self.Sill	Wert des Variogrammparameters Sill
self.Range	Wert des Variogrammparameters Range
self.minLat	Minimale Latitude (-90°)
self.maxLat	Maximale Latitude (+90°)
self.minLon	Minimale Longitude (-180°)
self.maxLon	Maximale Longitude (+180°)
self.Resolution	Wie in Kapitel 2.2.1
self.Outputfolder	Wie in Kapitel 2.2.1
self.Model	Hintergrundmodell („nequick“, „npdm“ oder „nphm“ [keine Doppelnennungen])
self.Layer	„NmF2“ oder „hmF2“ (keine Doppelnennungen)
self.VarMod	Variogrammmodelle „gau“, „pow“, „exp“ oder „sph“ (keine Doppelnennungen)
self.DoValidation	Wie in Kapitel 2.2.1
self.F107	Wie in Kapitel 2.2.1
self.Hour	Wie in Kapitel 2.2.1
self.Month	Wie in Kapitel 2.2.1
self.Doy	Wie in Kapitel 2.2.1
self.Lag	Wie in Kapitel 2.2.1

self.OutputfolderLog	Pfad für die Log Dateien
self.Filterdistance	Distanz, bei der die Messstationen für das Variogramm aussortiert werden.

## Methoden

__init__(stations, variogrammodell, outputfolder, model, doValidation, f107, hour, doy, month, resolution=2.5)	<p>Konstruktor.</p> <p><u>Übergabeparameter:</u></p> <p>stations: entspricht <i>self.Stations</i></p> <p>variogrammodell: entspricht <i>self.VarMod</i></p> <p>outputfolder: entspricht <i>self.Outputfolder</i></p> <p>model: entspricht <i>self.Model</i></p> <p>doValidation: entspricht <i>self.DoValidation</i></p> <p>f107: entspricht <i>self.F107</i></p> <p>hour: entspricht <i>self.Hour</i></p> <p>doy: entspricht <i>self.Doy</i></p> <p>month: entspricht <i>self.Month</i></p> <p>resolution: entspricht <i>self.Resolution</i></p>
myKriging()	Diese Funktion bestimmt global die Werte und den Fehler der Kriging Schätzung.
getModelValue(plat, plon)	<p>Diese Funktion berechnet den Modellwert (NeQuick, NPDM oder NPHM) an der Position (plat / plon). Dieser Wert wird für die Kriging Schätzung benötigt.</p> <p><u>Übergabeparameter:</u></p> <p>plat: Latitude</p> <p>plon: Longitude</p> <p><u>Return:</u></p> <p>Modellwert</p>
spatialDistanceVector (lats, lons, lat2, lon2)	<p>Diese Methode berechnet einen Vektor, der die sphärischen Distanzen zwischen allen Latituden und Longituden aus lats und lons und lat2 und lon2 beinhaltet.</p> <p><u>Übergabeparameter:</u></p> <p>lats: Liste aller Latituden</p>

	<p>lons: Liste aller Longituden</p> <p>lat2: Latitude</p> <p>lon2: Longitude</p> <p><u>Return:</u></p> <p>Liste aller Distanzen</p>
getInverse(coords)	<p>Diese Funktion berechnet die Inverse Matrix</p> <p><u>Übergabeparameter:</u></p> <p>coords: Liste aller Koordinaten (lat, lon)</p> <p><u>Return:</u></p> <p>Inverse Matrix</p>
getGammasMatrix (distmatrix)	<p>Diese Funktion berechnet die Gamma Matrix. Die Gamma Werte sind die Ergebnisse der Variogrammfunktion.</p> <p><u>Übergabeparameter:</u></p> <p>distmatrix: Distanzmatrix, die alle Distanzen zwischen den Messstationen beinhaltet.</p> <p><u>Return:</u></p> <p>Matrix mit allen Variogrammfunktionswerten zu den Distanzen.</p>
spatialDistanceMatrix (coordsRow, coordsColumn)	<p>Diese Funktion ist der Funktion <i>spatialDistanceVector(...)</i> ähnlich, nur das hier kein Vektor, sondern eine Matrix angegeben ist.</p> <p><u>Übergabeparameter:</u></p> <p>coordsRow: Koordinaten jeder Messstationen</p> <p>coordsColumn: Koordinaten jeder Messstationen (gleich mit coordsRow)</p> <p><u>Return:</u></p> <p>Matrix mit den Distanzen</p>
getGammasVector (distvector)	<p>Diese Funktion berechnet die Variogrammwerte für alle Distanzen in der übergebenen Liste.</p> <p><u>Übergabeparameter:</u></p> <p>distvector: Liste mit Distanzen</p> <p><u>Return:</u></p> <p>Liste mit allen Gammawerten</p>

<p>calcDistSpherical (lat1, lon1, lat2, lon2)</p>	<p>Diese Funktion berechnet die sphärische Distanz zwischen (lat1 / lon1) und (lat2 / lon2).</p> <p><u>Übergabeparameter:</u></p> <p>lat1: Latitude der Startposition lon1: Longitude der Startposition lat2: Latitude der Endposition lon2: Longitude der Endposition</p> <p><u>Return:</u></p> <p>Sphärische Distanz</p>
<p>plotKrigePrediction (valueToPlot, title)</p>	<p>Die Funktion stellt die Krige Schätzungen graphisch dar.</p> <p><u>Übergabeparameter:</u></p> <p>valueToPlot: Matrix; Kriging Schätzungen title: Titel der Abbildung</p>
<p>plotPredError (valueToPlot, title)</p>	<p>Die Funktion stellt den Schätzfehler graphisch dar.</p> <p><u>Übergabeparameter:</u></p> <p>valueToPlot: Matrix; Schätzfehler title: Titel der Abbildung</p>
<p>calcVariogramParams (lag)</p>	<p>Diese Funktion verwaltet die Variogrammschätzung. Ob diese in R oder in Python durchgeführt werden soll. Des Weiteren ruft sie die Plot Funktion für das Variogramm auf.</p> <p><u>Übergabeparameter:</u></p> <p>lag: entspricht <i>self.Lag</i></p>
<p>filterDistSemivar (distances, semivar)</p>	<p>Diese Funktion entfernt die Varianzen von Messstationen, die eine große Distanz als 10.000km zueinander haben. Dadurch wird die Variogrammschätzung durch die Messstationen nicht beeinflusst.</p> <p><u>Übergabeparameter:</u></p> <p>distances: Liste aller Distanzen semivar: Liste aller Semivarianzen</p> <p><u>Return:</u></p> <p>Liste aller Distanzen kleiner als 10.000km, Liste der dazugehörigen Semivarianzen</p>

<p>leastSquareFit(param, semivar, distances)</p>	<p>Diese Funktion schätzt die Variogrammparameter mittels der Methode der kleinsten Quadrate.</p> <p><u>Übergabeparameter:</u></p> <p>param: Liste der Variogrammparameter mit deren Initialwerten (sill, range, nugget)</p> <p>semivar: Liste der Semivarianzen</p> <p>distances: Liste der Distanzen</p> <p><u>Return:</u></p> <p>Liste der geschätzten Variogrammparameter (Sill, Range, Nugget)</p>
<p>plotVariogram (distances, semivar, varioparams)</p>	<p>Diese Funktion stellt sowohl die Variogrammwolke mit der Variogrammkurve als auch das experimentelle Variogramm mit Variogrammkurve dar.</p> <p><u>Übergabeparameter:</u></p> <p>distances: Liste der Distanzen</p> <p>semivar: Liste der Semivarianzen</p> <p>varioparams: Liste der Variogrammparameter (sill, range, nugget)</p>
<p>getMeanInLag (distVal, lagIntervall)</p>	<p>Diese Funktion bestimmt die Mittelwerte der Varianzen innerhalb eines Lags.</p> <p><u>Übergabeparameter:</u></p> <p>distVal: Gezippte Liste der Distanzen und der Varianzen</p> <p>lagIntervall: Liste der Lagintervalle</p> <p><u>Return:</u></p> <p>Liste der berechneten Mittelwerte</p>
<p>calcMean(list)</p>	<p>Berechnet den Mittelwert aus den Werten in der Liste.</p> <p><u>Übergabeparameter:</u></p> <p>list: Liste mit Werten</p> <p><u>Return:</u></p> <p>Mittelwert</p>
<p>gaussian(h)</p>	<p>Berechnet den Variogrammwert der Gausschen Variogrammfunktion für den Abstand <math>h</math>.</p> <p><u>Übergabeparameter:</u></p>

	<p>h: Distanz</p> <p><u>Return:</u></p> <p>Variogrammwert</p>
spherical(h)	<p>Berechnet den Variogrammwert der sphärischen Variogrammfunktion für den Abstand <math>h</math>.</p> <p><u>Übergabeparameter:</u></p> <p>h: Distanz</p> <p><u>Return:</u></p> <p>Variogrammwert</p>
exponential(h)	<p>Berechnet den Variogrammwert der exponentiellen Variogrammfunktion für den Abstand <math>h</math>.</p> <p><u>Übergabeparameter:</u></p> <p>h: Distanz</p> <p><u>Return:</u></p> <p>Variogrammwert</p>
power(h)	<p>Berechnet den Variogrammwert der Power Variogrammfunktion für den Abstand <math>h</math>.</p> <p><u>Übergabeparameter:</u></p> <p>h: Distanz</p> <p><u>Return:</u></p> <p>Variogrammwert</p>
writeInListofResults (filename)	<p>Wie in Kapitel 2.2.1</p>
findMinMax(matrix)	<p>Diese Funktion findet den minimalen und maximalen Wert in einer Matrix.</p> <p><u>Übergabeparameter:</u></p> <p>matrix: Matrix mit Werten</p> <p><u>Return:</u></p> <p>Minimum, Maximum</p>
manageEstimation LeastSquare (distances, semivar)	<p>Diese Funktion schätzt die Variogrammparameter mit der Methode der kleinsten Quadrate.</p> <p><u>Übergabeparameter:</u></p> <p>distances: Liste der Distanzen</p>

	semivar: Liste der Semivarianzen
writeResiduals4Validation(residual, stationid, filename)	Schreibt die Residuen der Referenzstationen in eine Datei <u>Übergabeparameter:</u> residual: Liste der Residuen stationid: Stationsid filename: Daateiname
writeInLog(message)	Schreibt in die Log Datei. <u>Übergabeparameter:</u> message: Zu schreibende Inhalt
manageEstimationAnisotropy(distances, semivar, angles)	Diese Funktion berechnet die Variogramme richtungsabhängig (Nord-Süd und Ost-West). <u>Übergabeparameter:</u> distances: Liste der Distanzen semivar: Liste der Semivarianzen angles: Liste der Winkel zwischen den Stationen
getGammasVectorAnisotropy(distvector)	Diese Funktion berechnet die Variogrammwerte für alle Distanzen in der übergebenen Liste. <u>Übergabeparameter:</u> distvector: Liste mit Distanzen <u>Return:</u> Liste mit allen Gammawerten
getGammasMatrixAnisotropy(distmatrix)	Diese Funktion berechnet die Gamma Matrix. Die Gamma Werte sind die Ergebnisse der Variogrammfunktion. <u>Übergabeparameter:</u> distmatrix: Distanzmatrix, die alle Distanzen zwischen den Messstationen beinhaltet. <u>Return:</u> Matrix mit allen Variogrammfunktionswerten zu den Distanzen.

### 2.1.3 MeasurementStations.py

Die Klasse MeasurementStations beschreibt ein Objekt für die Messstationen und deren Werte. Des Weiteren werden „getter“ und „setter“ Methoden zur Verfügung gestellt, um auf die Attribute zuzugreifen.

#### Globale Variablen

self.Lat	Liste der Latituden der Messstationen
self.Lon	Liste der Longituden der Messstationen
self.MeasurementValue	Liste der Messwerte der Messstationen
self.DoubleStation	Boolscher Wert, der angibt, ob Messstationen doppelt in den Listen vorkommen (True) oder nicht (False).
self.StationID	Liste der IDs der Messstationen
self.Residuals	Liste der Residuen
self.Layer	„NmF2“ oder „hmF2“ (keine Doppelnennungen)
self.RefStations	Liste der IDs der Referenzstationen

#### Methoden

__init__(doubleStations, layer, refstations=[])	Konstruktor. <u>Übergabeparameter:</u> doubleStations: entspricht <i>self.DoubleStation</i> layer: entspricht <i>self.Layer</i> refstations: entspricht <i>self.RefStations</i> , muss aber nicht unbedingt angegeben werden.
addStation(lat, lon, measurement, stationid, residual=-9999)	Diese Funktion fügt eine Messstation hinzu. <u>Übergabeparameter:</u> lat: Latitude der Messstation lon: Longitude der Messstation measurement: Messwert der Messstation stationid: Messstation ID residual: Residuum, wenn nicht angegeben, wird er nicht berücksichtigt.

getStation(position)	<p>Diese Funktion gibt die Koordinaten und den Messwert einer Messstation zurück.</p> <p><u>Übergabeparameter:</u> position: Position in der Liste der Messstationen</p> <p><u>Return:</u> Latitude, Longitude, Messwert</p>
getLat(position)	<p>Diese Funktion gibt die Latitude einer Messstation zurück.</p> <p><u>Übergabeparameter:</u> position: Position in der Liste der Messstationen</p> <p><u>Return:</u> Latitude</p>
getLon(position)	<p>Diese Funktion gibt die Longitude einer Messstation zurück.</p> <p><u>Übergabeparameter:</u> position: Position in der Liste der Messstationen</p> <p><u>Return:</u> Longitude</p>
getAllStationID	<p>Diese Funktion gibt die komplette Liste der Station IDs zurück.</p> <p><u>Return:</u> Liste der Station IDs</p>
getStationID(position)	<p>Diese Funktion gibt die Stations ID einer Messstation zurück.</p> <p><u>Übergabeparameter:</u> position: Position in der Liste der Messstationen</p> <p><u>Return:</u> Stations ID</p>
getMeasurement(position)	<p>Diese Funktion gibt den Messwert einer Messstation zurück.</p> <p><u>Übergabeparameter:</u> position: Position in der Liste der Messstationen</p> <p><u>Return:</u></p>

	Messwert
getAllStation()	Diese Funktion gibt die Koordinaten und die Messwerte aller Messstation zurück. <u>Return:</u> Liste der Latituden, Liste der Longituden, Liste der Messwerte
getAllLat()	Diese Funktion gibt die komplette Liste der Latituden zurück. <u>Return:</u> Liste aller Latituden
getAllLon()	Diese Funktion gibt die komplette Liste der Longituden zurück. <u>Return:</u> Liste aller Longituden
getAllMeasurement()	Diese Funktion gibt die komplette Liste der Messwerte zurück. <u>Return:</u> Liste aller Messwerte
getMean()	Diese Funktion gibt den Mittelwert der Messwerte zurück. <u>Return:</u> Mittelwert
getCount()	Diese Funktion gibt die Anzahl der Messstationen zurück. <u>Return:</u> Anzahl der Messstationen
deleteOneStation(lat, lon)	Diese Funktion löscht eine Messstation bei den angegebenen Koordinaten. <u>Übergabeparameter:</u> lat: Latitude der zu löschenden Messstation lon: Longitude der zu löschenden Messstation
calcfof2TONmF2()	Diese Funktion rechnet foF2-Werte in NmF2-Werte um.

getMeanOneStation(lat, lon)	<p>Diese Funktion berechnet den Mittelwert von Werten einer Messstation.</p> <p><u>Übergabeparameter:</u></p> <p>lat: Latitude der Messstation lon: Longitude der Messstation</p> <p><u>Return:</u></p> <p>Mittelwert</p>
diffModel(modelValues)	<p>Diese Funktion berechnet die Residuen (Differenz zwischen Messwert und Modellwert)</p> <p><u>Übergabeparameter:</u></p> <p>modelValues: Liste der Modellwerte</p> <p><u>Return:</u></p> <p>Liste der Residuen</p>
diffModelRelative(modelValues)	<p>Diese Funktion berechnet die relativen Differenzen zwischen Messwert und Modellwert.</p> <p><u>Übergabeparameter:</u></p> <p>modelValues: Liste der Modellwerte</p> <p><u>Return:</u></p> <p>Liste der relativen Distanzen</p>
removeRefStations()	<p>Diese Funktion erstellt ein neues <i>MeasurementStation</i>-Objekt ohne die Referenzstationen.</p> <p><u>Return:</u></p> <p><i>MeasurementStation</i>-Objekt</p>
addResiduals(residuals)	<p>Diese Funktion fügt eine Liste von Residuen dem Attribut <i>self.Residuals</i> hinzu.</p> <p><u>Übergabeparameter:</u></p> <p>residuals: Liste der Residuen</p>
getResiduals()	<p>Diese Funktion gibt die Liste der Residuen zurück.</p> <p><u>Return:</u></p> <p>Liste der Residuen</p>
getLayer()	<p>Diese Funktion gibt den Layer, „NmF2“ oder „hmF2“ zurück.</p> <p><u>Return:</u></p>

	Layer
getRefStations()	Diese Funktion erstellt ein neues <i>MeasurementStation</i> -Objekt, das nur die Referenzstationen beinhaltet. <u>Return:</u> <i>MeasurementStation</i> -Objekt
calcDistSpherical (lat1, lon1, lat2, lon2)	Diese Funktion berechnet die sphärische Distanz zwischen (lat1 / lon1) und (lat2 / lon2). <u>Übergabeparameter:</u> lat1: Latitude der Startposition lon1: Longitude der Startposition lat2: Latitude der Endposition lon2: Longitude der Endposition <u>Return:</u> Sphärische Distanz

#### 2.1.4 KrigeValidation.py

Diese Klasse stellt Methoden zur Verfügung, um die Ergebnisse des Kriging Moduls mit anderen Ergebnissen von anderen Schätzmethoden zu vergleichen. Dieser Vergleich ist optional.

#### Globale Variablen

self.PredictionMatrix	Die vom Kriging Modul geschätzten Werte als Matrix. Die Zeilen entsprechen der Latituden, die Spalten der Longituden.
self.ReadMatrix	Vergleichsmatrix, die Werte aus anderen Schätzverfahren beinhaltet. Die Zeilen entsprechen der Latituden, die Spalten der Longituden.
self.Header	Kopfzeilen für die ASCII-Datei, die die Differenzen zwischen <i>self.PredictionMatrix</i> und <i>self.ReadMatrix</i> beinhaltet.
self.Path	Entspricht den <i>self.Outputfolder</i> aus Kapitel 2.2.1
self.Layer	Wie in Kapitel 2.2.1

self.VarModel	Wie in Kapitel 2.2.1
self.PathComp	Entspricht den <i>self.PathToCompFilehmF2</i> oder <i>self.PathToCompFileNmF2</i> aus Kapitel 2.2.1 je nach Layer.
self.BackModel	Wie in Kapitel 2.2.1

## Methoden

<code>__init__(predictionMatrix, header, path, layer, varmodel, pathCompFile, usedModel)</code>	<p>Konstruktor.</p> <p><u>Übergabeparameter:</u></p> <p>predictionMatrix: entspricht <i>self.PredictionMatrix</i>  header: entspricht <i>self.Header</i>  path: entspricht <i>self.Path</i>  layer: entspricht <i>self.Layer</i>  varmodel: entspricht <i>self.VarModel</i>  pathCompFile: entspricht <i>self.PathComp</i>  usedModel: entspricht <i>self.BackModel</i></p>
<code>doValidation()</code>	Diese Funktion verwaltet die Validierung mit anderen Ergebnissen. Diese Funktion sollte ebenfalls aufgerufen werden, um die Validierung zu starten, nach der Objektinitialisierung.
<code>readMatrix()</code>	Diese Funktion liest die Datei, angegeben in <i>self.PathComp</i> aus.
<code>printMatrix(filename)</code>	<p>Diese Funktion schreibt die <i>self.predictionMatrix</i> in eine ASCII-Datei.</p> <p><u>Übergabeparameter:</u></p> <p>filename: Dateiname</p>
<code>printDiffAbs(filename)</code>	<p>Diese Funktion schreibt die absoluten Differenzen zwischen <i>self.PredictionMatrix</i> und <i>self.ReadMatrix</i> in eine ASCII-Datei.</p> <p><u>Übergabeparameter:</u></p> <p>filename: Dateiname</p>

printDiffRel(filename)	<p>Diese Funktion schreibt die relativen Differenzen zwischen <i>self.PredictionMatrix</i> und <i>self.ReadMatrix</i> in eine ASCII-Datei.</p> <p><u>Übergabeparameter:</u> filename: Dateiname</p>
plotDiff(difference, filename)	<p>Diese Funktion plottet eine Weltkarte, die die übergebenen Differenzen farblich darstellt.</p> <p><u>Übergabeparameter:</u> difference: Differenzen zwischen <i>self.PredictionMatrix</i> und <i>self.ReadMatrix</i> filename: Dateiname</p>
calcMeanAndStdMatrix (matrix)	<p>Diese Funktion berechnet den Mittelwert und die Standardabweichung von Werten in einer Matrix.</p> <p><u>Übergabeparameter:</u> matrix: Matrix mit Werten</p> <p><u>Return:</u> Mittelwert, Standardabweichung</p>
findMinMax(matrix)	<p>Diese Funktion berechnet den minimalen und maximalen Wert in einer Matrix.</p> <p><u>Übergabeparameter:</u> matrix: Matrix mit Werten</p> <p><u>Return:</u> Minimum, Maximum</p>
writelnListofResults (filename)	Wie in Kapitel 2.2.1
plotHistMeasurementEstimation(filename, _xlabel, _ylabel)	<p>Diese Funktion stellt die Residuen zwischen Mess- und Schätzwert in einem Histogramm dar.</p> <p><u>Übergabeparameter:</u> filename: Dateiname mit den Residuen _xlabel: Name der X-Achse des Histogramms _ylabel: Name der Y-Achse des Histogramms</p>

plotHist4RefStation(filename, _xlabel, _ylabel)	<p>Diese Funktion gibt die Residuen zwischen Mess- und Schätzwert in einem Histogramm für jede einzelne Referenzstation aus.</p> <p><u>Übergabeparameter:</u></p> <p>filename: Dateiname mit den Residuen  _xlabel: Name der X-Achse des Hisograms  _ylabel: Name der Y-Achse des Hisograms</p>
---	---

### 2.1.5 VariogramInR.py

Diese Klasse verwaltet die Variogrammberechnung in R. Sie wird nur verwendet, wenn das Packet „rpy2“ installiert ist. Ist das nicht der Fall, erfolgt die Variogrammschätzung in Python.

#### Globale Variablen

self.Layer	„NmF2“ oder „hmF2“ (keine Doppelnennungen)
self.Lag	Lag Distanz
self.Filterdistance	Filterdistance für das Variogramm
self.VarMod	Variogrammmodell
self.Outputfolder	Zielpfad für die Ausgabedateien
self.OutputfolderLog	Zielpfad für die Log Dateien
self.Model	Hintergrundmodell
self.WithStructure	Legt fest, ob R eine Variogrammstruktur vorgegeben wird oder nicht.

#### Methoden

__init__(self, lag, filterdistance, varmod, layer, outputfolder, outputfolderlog, model)	<p>Konstruktor.</p> <p><u>Übergabeparameter:</u></p> <p>lag: Lag Distanz  filterdistance: Filterdistance für das Variogramm  varmod: Variogrammmodell</p>
--	---

	<p>layer: NmF2 oder hmF2</p> <p>outputfolder: Zielpfad für die Ausgabedateien</p> <p>outputfolderlog: Zielpfad für die Log Dateien</p> <p>model: Hintergrundmodell</p>
<p>manageEstimation InR(distances, semivar)</p>	<p>Diese Funktion verwaltet die Schätzung der Variogrammparameter in R.</p> <p><u>Übergabeparameter:</u></p> <p>distances: Liste der Distanzen</p> <p>semivar: Liste der Semivarianzen</p>
<p>writeTemp4R (lagIntervall, var, counterofLag)</p>	<p>Diese Funktion erstellt eine temporäre (CSV) Datei, die die Übergabeparameter beinhaltet. Sie wird benötigt, um die Werte in die Programmiersprache R zu überführen.</p> <p><u>Übergabeparameter:</u></p> <p>lagIntervall: Liste der Lagintervalle</p> <p>var: Liste der Varianzen aus dem experimentellen Variogramm</p> <p>counterofLag: Liste, die die Anzahl der Varianzen in einem Lag beinhaltet.</p>
<p>deleteExchangeFile()</p>	<p>Diese Funktion löscht die durch die Funktion <i>writeTemp4R(...)</i> erstellte Datei.</p>
<p>estimateVarInR (amountOfLags)</p>	<p>Diese Funktion berechnet die Variogrammparameter in R.</p> <p><u>Übergabeparameter:</u></p> <p>amountOfLags: Anzahl der Lags</p>
<p>writeInListofResults(filename)</p>	<p>Diese Funktion schreibt Dateinamen in die „ListofResults.dat“, die alle Dateinamen von allen generierten Dateien enthält.</p> <p><u>Übergabeparameter:</u></p> <p>filename: Dateiname</p>
<p>gaussian(h)</p>	<p>Berechnet den Variogrammwert der Gausschen Variogrammfunktion für den Abstand <math>h</math>.</p> <p><u>Übergabeparameter:</u></p> <p>h: Distanz</p> <p><u>Return:</u></p>

	Variogrammwert
spherical(h)	Berechnet den Variogrammwert der sphärischen Variogrammfunktion für den Abstand $h$ . <u>Übergabeparameter:</u> h: Distanz <u>Return:</u> Variogrammwert
exponential(h)	Berechnet den Variogrammwert der exponentiellen Variogrammfunktion für den Abstand $h$ . <u>Übergabeparameter:</u> h: Distanz <u>Return:</u> Variogrammwert
power(h)	Berechnet den Variogrammwert der Power Variogrammfunktion für den Abstand $h$ . <u>Übergabeparameter:</u> h: Distanz <u>Return:</u> Variogrammwert
plotVariogram (distances, semivar, varioparams)	Diese Funktion stellt sowohl die Variogrammwolke mit der Variogrammkurve als auch das experimentelle Variogramm mit Variogrammkurve dar. <u>Übergabeparameter:</u> distances: Liste der Distanzen semivar: Liste der Semivarianzen varioparams: Liste der Variogrammparameter (sill, range, nugget)