



# Master Thesis

im Rahmen des

Universitätslehrganges „Geographical Information Science & Systems“  
(UNIGIS MSc) am Interfakultären Fachbereich für GeoInformatik (Z\_GIS)  
der Paris Lodron-Universität Salzburg

zum Thema

## Vergleich gängiger Interpolationsalgorithmen zur digitalen Oberflächenmodellierung

vorgelegt von

Dipl.-HTL-Ing.

**Erwin Winkler**

10535, UNIGIS MSc Jahrgang 2018

Zur Erlangung des Grades

„Master of Science (Geographical Information Science & Systems) – MSc(GIS)“

Nußdorf, 01.02.2020

## Danksagung

Zuerst möchte ich mich beim gesamten Lehrgangsteam der Universität Salzburg für die hervorragende Betreuung im Rahmen meines UNIGIS Professional und späteren UNIGIS MSc Studiums bedanken. Die Kommunikation mit dem Lehrgangsbüro und mit allen Modul-Betreuern fand stets auf freundschaftlichem Niveau statt, und meine Fragen wurden immer rasch und zu meiner vollsten Zufriedenheit beantwortet.

Des Weiteren möchte ich mich besonders bei meinem MT-Betreuer Herrn Dr. Christian Neuwirth bedanken, der mich bereits während der Themenfindung und der späteren Erstellung dieser Arbeit sehr unterstützt hat und mir dabei geholfen hat, trotz eines schweren Freizeitemfalls im letzten Jahr, meine Motivation wiederzufinden um das Studium doch noch erfolgreich abschließen zu können.

Abschließend gilt mein innigster und größter Dank meinen Eltern Erwin und Elisabeth Winkler, welche mich nicht nur bei diesem Studium immens unterstützt haben, sondern mir im gesamten Verlauf meines Lebens immer mit Rat und Tat zur Seite standen, und somit auch maßgeblich zum Gelingen dieser Master Thesis beigetragen haben.

Vielen Dank!

## Eigenständigkeitserklärung

„Ich versichere, die vorliegende Master Thesis ohne fremde Hilfe und ohne Verwendung anderer als der angeführten Quellen angefertigt zu haben, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen ist. Alle Ausführungen der Arbeit, die wörtlich oder sinngemäß übernommen wurden, sind entsprechend gekennzeichnet.“

Nußdorf, 01. Februar 2020

---

Ort, Datum

Erwin Winkler

## Zusammenfassung

Interpolationsalgorithmen gehören zu den am häufigst verwendeten (Geo-)Algorithmen im Zusammenhang mit geographischen Informationssystemen. Mittels dieser Algorithmen werden aus Vektordaten, in Form von Punkterastern als Eingangsdaten(quellen), digitale Oberflächenmodelle erzeugt, wobei hierfür grundsätzlich unterschiedliche Interpolationsverfahren zur Verfügung stehen. Diese können prinzipiell in zwei große Gruppen unterteilt werden: einerseits in die deterministischen und andererseits in die probabilistischen Interpolationsverfahren. Zur ersten Gruppe gehören u.a. die „*Inverse Distance Weighted*“-Interpolation, die „*Global Polynomial Interpolation*“, die „*Local Polynomial Interpolation*“, die „*Natural Neighbor*“-Methode sowie die „*Spline*“-Interpolation. Die Berechnung der Interpolationsoberflächen erfolgt dabei rein anhand mathematischer Funktionen und es werden weiters keine potentiell möglichen räumlichen Autokorrelationen zwischen den (Mess-)Punkten berücksichtigt. Bei der zweiten Gruppe handelt sich um eine Kombination mathematischer und statistischer Verfahren, wobei die Ermittlung der Schätzwerte auf Basis von Wahrscheinlichkeiten beruht. Es werden neben den Distanzen zwischen den (Mess-)Punkten und den Schätzwerten auch die räumlichen Autokorrelationen der (Mess-)Punkte untereinander mitberücksichtigt. Man spricht hier von sogenannten geostatistischen Interpolationsverfahren. Das wohl bekannteste Verfahren dieser Gruppe ist der „*Kriging*“-Interpolationsalgorithmus.

Das Ziel der vorliegenden Master Thesis ist der wissenschaftliche Vergleich der drei am häufigst verwendeten Interpolationsalgorithmen bzw. Verfahren: „*Inverse Distance Weighted*“, „*Kriging*“ und „*Spline*“. Hierfür werden anhand eines automatisierten Geoprozesses in Form eines Python Add-Ins für ESRI's ArcGIS Pro, und insgesamt drei verschiedenen Basisdatensätzen als Eingangsdaten, jeweils mehrere Interpolationsoberflächen je Verfahren erzeugt, wodurch u.a. die Sensitivität der einzelnen Interpolationsverfahren überprüft wird. Anschließend erfolgt eine visuelle Analyse, eine Explorative Räumliche Datenanalyse („*ESDA*“) und eine Kreuzvalidierung der daraus resultierenden Interpolationsergebnisse. Die Validierung erfolgt dabei auf Basis der Ermittlung folgender Fehlerkennzahlen: dem „*Mean Error*“ („*ME*“) und dem „*Root Mean Square Error*“ („*RMSE*“), wobei diese für alle Interpolationsoberflächen berechnet werden. Die Kreuzvalidierung auf Basis der drei für den vorliegenden Vergleich verwendeten Basisdatensätze zeigt, dass grundsätzlich der „*Spline*“-Interpolationsalgorithmus jeweils die geringsten Fehlerwerte („*ME*“ sowie „*RMSE*“) generiert und somit letztlich die besten Interpolationsresultate liefert.

Schlagwörter: *IDW, Kriging, Spline, Root Mean Square Error, Kreuzvalidierung*

## Abstract

Interpolation algorithms are one of the most frequently used (geo)algorithms in combination with geographic information systems. Using these algorithms, digital surface models are generated from vector data in form of point grids as input data (source data), whereby fundamentally different interpolation methods are available for this purpose. In principle, these can be divided into two large groups: the deterministic and the probabilistic interpolation methods. The first group includes among others the inverse distance weighted interpolation, the global polynomial interpolation, the local polynomial interpolation, the natural neighbor method and the spline interpolation. The calculation of the interpolation surfaces is based purely on mathematical functions and no potentially possible spatial autocorrelations between the (measurement) points are considered. The second group is a combination of mathematical and statistical methods, whereby the determination of estimated values is based on probabilities. In addition to the distances between the (measurement) points and the estimated values, the spatial autocorrelations between the points are taken into account. These are also known as geostatistical interpolation methods. The probably best-known method of this second group is the kriging interpolation.

The aim of the present master thesis is the scientific comparison of the three most frequently used interpolation algorithms respectively procedures: inverse distance weighted, kriging, and spline. For this purpose, an automated geoprocess in form of a Python Add-In for ESRI's ArcGIS Pro, and a total of three different base data sets as input data, generate several interpolation surfaces per method, which, among other things, increases the sensitivity of the individual interpolation procedures is checked. This is followed by a visual analysis, an exploratory spatial data analysis (ESDA) and a cross-validation of the interpolation results. The validation is based on the determination of the following errors: the mean error (ME) and the root mean square error (RMSE), whereby these are calculated for all interpolation surfaces. The cross-validation on the basis of the three basic data sets used for the present comparison shows that the spline interpolation algorithm generates the lowest error values (ME and RMSE) and thus ultimately provides the best interpolation results.

Keywords: *IDW, Kriging, Spline, Root Mean Square Error, Cross-Validation*

## Inhaltsverzeichnis

Danksagung .....	II
Eigenständigkeitserklärung .....	III
Zusammenfassung .....	IV
Abstract .....	V
Inhaltsverzeichnis .....	VI
Abbildungsverzeichnis .....	VIII
Tabellenverzeichnis .....	X
Abkürzungsverzeichnis .....	XI
1. Einleitung .....	1
1.1 Themenfeld und Relevanz .....	1
1.2 Literaturüberblick .....	3
1.3 Zielsetzung und Forschungsfrage .....	5
1.4 Abgrenzung.....	7
1.5 Aufbau und Struktur der Arbeit.....	9
2. Theoretische Grundlagen.....	10
2.1 Grundlagen und historische Entwicklung von Interpolationsalgorithmen .....	10
2.2 Interpolationsalgorithmen.....	12
2.2.1 „INVERSE DISTANCE WEIGHTED“ („IDW“).....	12
2.2.2 „KRIGING“ .....	16
2.2.3 „SPLINE“ .....	23
2.3 Sonstige für den Geoprozess verwendete (Geo-)Algorithmen .....	27
3. Basisdatensätze .....	28
3.1 Datengrundlage.....	28
3.2 Basisdatensatz 1 .....	30

3.3	Basisdatensatz 2 .....	32
3.4	Basisdatensatz 3 .....	34
3.5	Zusammengefasste Charakteristika der drei Basisdatensätze .....	37
4.	Geoprozess .....	38
4.1	Spezifikation und Anforderungen .....	38
4.2	Sensitivitätsanalyse .....	41
4.3	Erstellung des Python Add-Ins .....	41
4.4	Implementierung .....	42
5.	Interpolationsergebnisse .....	42
5.1	Visuelle Darstellung der Ergebnisse .....	42
5.1.1	Ergebnisse – Basisdatensatz 1 .....	43
5.1.2	Ergebnisse – Basisdatensatz 2 .....	44
5.1.3	Ergebnisse – Basisdatensatz 3 .....	46
5.2	Explorative Räumliche Datenanalyse (ESDA) .....	48
5.3	Kreuzvalidierung .....	51
5.3.1	Ergebnisse – Basisdatensatz 1 .....	55
5.3.2	Ergebnisse – Basisdatensatz 2 .....	59
5.3.3	Ergebnisse – Basisdatensatz 3 .....	63
6.	Diskussion .....	67
7.	Schlussfolgerungen und Ausblick .....	71
	Literaturverzeichnis .....	73
	Anhang A – ArcGIS Pro Python Add-In .....	XII
	Anhang B – Häufigkeitsverteilungen der Interpolationsergebnisse .....	XXVII
	Anhang C – Referenzpunkte – berechnete Interpolationswerte .....	XXXVI
	Anhang D – „Nearest Neighbor“-Analysen der drei Basisdatensätze .....	XLV

## Abbildungsverzeichnis

Abbildung 1: Beispiel für ein digitales Geländemodell (DGM) .....	2
Abbildung 2: Interpolation einer Höhenoberfläche auf Basis von bekannten (Mess-)Punkten .....	10
Abbildung 3: IDW-Nachbarschaft für einen definierten Punkt.....	12
Abbildung 4: Beispiel 1 – resultierende Interpolationsfehler.....	14
Abbildung 5: Beispiel 2 – resultierende Interpolationsfehler.....	14
Abbildung 6: Interpolationsergebnisse.....	15
Abbildung 7: Beispiel für die Ermittlung von $Z(s_p)$ .....	18
Abbildung 8: Semivariogramm-Modell – „Spherical“ .....	19
Abbildung 9: Semivariogramm-Modell – „Circular“ .....	19
Abbildung 10: Semivariogramm-Modell – „Exponential“ .....	19
Abbildung 11: Semivariogramm-Modell – „Gaussian“ .....	19
Abbildung 12: Semivariogramm-Modell – „Linear“ .....	20
Abbildung 13: Visuelle Ergebnisse unterschiedlicher Interpolationsalgorithmen .....	21
Abbildung 14: Beispiel einer Kreuzvalidierung - „Ordinary-Kriging“-Variante (Bild 1/2).....	21
Abbildung 15: Beispiel einer Kreuzvalidierung - „Ordinary-Kriging“-Variante (Bild 2/2).....	22
Abbildung 16: „Regularized-Spline“-Interpolationsoberfläche mit „Weight“ = 0,5 .....	25
Abbildung 17: „Tension-Spline“-Interpolationsoberfläche mit „Weight“ = 5,0 .....	26
Abbildung 18: ÖK50 mit Detail-Orthofoto – Grenzen der Basisdaten (grüne umrandet) .....	28
Abbildung 19: Detail-Orthofoto – Räumliche Ausdehnung / Grenzen der BEV-Basisdaten .....	29
Abbildung 20: Trendoberfläche – Basisdatensatz 1 .....	30
Abbildung 21: Deskriptive stat. Maße – Basisdatensatz 1 .....	31
Abbildung 22: „Elevation“-Höhenattribut Histogramm – Basisdatensatz 1 .....	31
Abbildung 23: Trendoberfläche – Basisdatensatz 2 .....	32
Abbildung 24: Deskriptive stat. Maße – Basisdatensatz 2 .....	33
Abbildung 25: „Elevation“-Höhenattribut Histogramm – Basisdatensatz 2 .....	33
Abbildung 26: Räumliche Ausdehnung / Grenzen des 3. Basisdatensatzes.....	34
Abbildung 27: Deskriptive stat. Maße – Basisdatensatz 3 .....	35
Abbildung 28: „Elevation“-Höhenattribut Histogramm – Basisdatensatz 3 .....	36
Abbildung 29: Trendoberfläche – Basisdatensatz 3 .....	36
Abbildung 30: Schematische Darstellung des Geoprozesses.....	39
Abbildung 31: Interpolationsergebnis – „IDW“ („Power“ = 2) – Basisdatensatz 1.....	43
Abbildung 32: Interpolationsergebnis – „Ordinary-Kriging“ („Expon.“) – Basisdatensatz 1.....	43
Abbildung 33: Interpolationsergebnis – „Spline“ („Weight“ = 0,1) – Basisdatensatz 1 .....	44
Abbildung 34: Interpolationsergebnis – „IDW“ („Power“ = 3) – Basisdatensatz 2.....	44
Abbildung 35: Interpolationsergebnis – „Ordinary-Kriging“ („Expon.“) – Basisdatensatz 2.....	45
Abbildung 36: Interpolationsergebnis – „Spline“ („Weight“ = 0,05) – Basisdatensatz 2 .....	45
Abbildung 37: Interpolationsergebnis – „IDW“ („Power“ = 3) – Basisdatensatz 3.....	46
Abbildung 38: Interpolationsergebnis – „Ordinary-Kriging“ („Expon.“) – Basisdatensatz 3.....	46
Abbildung 39: Interpolationsergebnis – „Spline“ („Weight“ = 0,5) – Basisdatensatz 3 .....	47

Abbildung 40: Räumliche Verteilung der Referenzpunkte → Basisdatensatz 1 und 2 .....	53
Abbildung 41: Referenzpunkte inklusive Basisdatensätze 1 und 2 („BEV“-Daten).....	53
Abbildung 42: Räumliche Verteilung der Referenzpunkte → Basisdatensatz 3 .....	54
Abbildung 43: Referenzpunkte inklusive Basisdatensatz 3 („SRTM“-Daten) .....	54
Abbildung 44: ermittelte „Mean Errors“ – Basisdatensatz 1 .....	56
Abbildung 45: ermittelte „Root Mean Square Errors“ – Basisdatensatz 1 .....	56
Abbildung 46: „IDW“-Sensitivität – Basisdatensatz 1 .....	57
Abbildung 47: „Spline“-Sensitivität – Basisdatensatz 1 .....	57
Abbildung 48: „Mean Error“ zu „Root Mean Square Error“ – Basisdatensatz 1 .....	58
Abbildung 49: ermittelte „Mean Errors“ – Basisdatensatz 2 .....	60
Abbildung 50: ermittelte „Root Mean Square Errors“ – Basisdatensatz 2 .....	60
Abbildung 51: „IDW“-Sensitivität – Basisdatensatz 2 .....	61
Abbildung 52: „Spline“-Sensitivität – Basisdatensatz 2 .....	61
Abbildung 53: „Mean Error“ zu „Root Mean Square Error“ – Basisdatensatz 2 .....	62
Abbildung 54: ermittelte „Mean Errors“ – Basisdatensatz 3 .....	64
Abbildung 55: ermittelte „Root Mean Square Errors“ – Basisdatensatz 3 .....	64
Abbildung 56: „IDW“-Sensitivität – Basisdatensatz 3 .....	65
Abbildung 57: „Spline“-Sensitivität – Basisdatensatz 3 .....	65
Abbildung 58: „Mean Error“ zu „Root Mean Square Error“ – Basisdatensatz 3 .....	66
Abbildung 59: Werteverteilung für „IDW“ („Power“ = 2) – Basisdatensatz 1 .....	XXVII
Abbildung 60: Werteverteilung für „Ordinary-Kriging“ („Expon.“) – Basisdatensatz 1 .....	XXVIII
Abbildung 61: Werteverteilung für „Spline“ („Weight“ = 0,1) – Basisdatensatz 1 .....	XXIX
Abbildung 62: Werteverteilung für „IDW“ („Power“ = 3) – Basisdatensatz 2 .....	XXX
Abbildung 63: Werteverteilung für „Ordinary-Kriging“ („Expon.“) – Basisdatensatz 2 .....	XXXI
Abbildung 64: Werteverteilung für „Spline“ („Weight“ = 0,05) – Basisdatensatz 2 .....	XXXII
Abbildung 65: Werteverteilung für „IDW“ („Power“ = 3) – Basisdatensatz 3 .....	XXXIII
Abbildung 66: Werteverteilung für „Ordinary-Kriging“ („Expon.“) – Basisdatensatz 3 .....	XXXIV
Abbildung 67: Werteverteilung für „Spline“ („Weight“ = 0,5) – Basisdatensatz 3 .....	XXXV
Abbildung 68: „Nearest Neighbor“-Analyse – Basisdatensatz 1 .....	XLV
Abbildung 69: „Nearest Neighbor“-Analyse – Basisdatensatz 2 .....	XLVI
Abbildung 70: „Nearest Neighbor“-Analyse – Basisdatensatz 3 .....	XLVII

## Tabellenverzeichnis

Tabelle 1: Abkürzungsverzeichnis .....	XI
Tabelle 2: Grundlegende Informationen – Basisdatensatz 1 .....	30
Tabelle 3: Grundlegende Informationen – Basisdatensatz 2 .....	32
Tabelle 4: Grundlegende Informationen – Basisdatensatz 3 .....	35
Tabelle 5: Zusammengefasste Charakteristika – Basisdatensatz 1 bis 3 .....	37
Tabelle 6: Vorgabewerte – „IDW“-Interpolationsalgorithmus .....	40
Tabelle 7: Vorgabewerte – „Kriging“-Interpolationsalgorithmus .....	40
Tabelle 8: Vorgabewerte – „Spline“-Interpolationsalgorithmus .....	40
Tabelle 9: Deskriptive statistische Maße aller erstellten Interpolationsoberflächen .....	50
Tabelle 10: Referenzpunkte – Basisdatensatz 1 bis 3 .....	52
Tabelle 11: Ergebnisse der Kreuzvalidierung („ME“ und „RMSE“) – Basisdatensatz 1 .....	55
Tabelle 12: Ergebnisse der Kreuzvalidierung („ME“ und „RMSE“) – Basisdatensatz 2 .....	59
Tabelle 13: Ergebnisse der Kreuzvalidierung („ME“ und „RMSE“) – Basisdatensatz 3 .....	63
Tabelle 14: Gegenüberstellung der ermittelten Fehlerwerte („ME“ bzw. „RMSE“)	71
Tabelle 15: „IDW“-Zellenwerte – Basisdatensatz 1 .....	XXXVI
Tabelle 16: „Kriging“-Zellenwerte – Basisdatensatz 1 .....	XXXVII
Tabelle 17: „Spline“-Zellenwerte – Basisdatensatz 1 .....	XXXVIII
Tabelle 18: „IDW“-Zellenwerte – Basisdatensatz 2 .....	XXXIX
Tabelle 19: „Kriging“-Zellenwerte – Basisdatensatz 2 .....	XL
Tabelle 20: „Spline“-Zellenwerte – Basisdatensatz 2 .....	XLI
Tabelle 21: „IDW“-Zellenwerte – Basisdatensatz 3 .....	XLII
Tabelle 22: „Kriging“-Zellenwerte – Basisdatensatz 3 .....	XLIII
Tabelle 23: „Spline“-Zellenwerte – Basisdatensatz 3 .....	XLIV

## Abkürzungsverzeichnis

2D	Zweidimensional
3D	Dreidimensional
ASE	<i>Average Standard Error</i>
BEV	Bundesamt für Eich- und Vermessungswesen
CAD	<i>Computer Aided Design</i>
DEM	<i>Digital Elevation Model</i>
DGM	Digitales Geländemodell
DXF	<i>Data Exchange Format</i>
EBK	<i>Empirical Bayesian Kriging</i>
ESDA	<i>Explorative Spatial Data Analysis</i>
ESRI	<i>Environmental Systems Research Institute</i>
ETRS89	<i>European Terrestrial Reference System 1989</i>
GIS	Geographisches Informationssystem
GPI	<i>Global Polynomial Interpolation</i>
GPS	<i>Global Positioning System</i>
IDW	<i>Inverse Distance Weighted</i>
IK	<i>Indicator Kriging</i>
LPI	<i>Local Polynomial Interpolation</i>
MAE	<i>Mean Absolute Error</i>
ME	<i>Mean Error</i>
MRE	<i>Mean Relative Error</i>
MSE	<i>Mean Standardized Error</i>
NN	<i>Natural Neighbor</i>
NNA	<i>Nearest Neighbor Analysis</i>
OK	<i>Ordinary Kriging</i>
RBF	<i>Radial Basis Function</i>
RMSE	<i>Root Mean Square Error</i>
RMSSE	<i>Root Mean Square Standardized Error</i>
SK	<i>Simple Kriging</i>
SRTM	<i>Shuttle Radar Topography Mission</i>
TIN	<i>Triangular Irregular Network</i>
UK	<i>Universal Kriging</i>
UTM	<i>Universal Transverse Mercator</i>
WEA(s)	Windenergieanlage(n)

Tabelle 1: Abkürzungsverzeichnis

# 1. Einleitung

## 1.1 Themenfeld und Relevanz

Die Bedeutung von geographischen Informationssystemen (GIS) hat in den letzten Jahrzehnten enorm zugenommen. Diese werden heutzutage in einer Vielzahl von Bereichen des alltäglichen Lebens eingesetzt. Die Anwendungsgebiete reichen von der Landschafts- und Stadtplanung, der Analyse von Verkehrs- und Transportrouten, der Planung von humanitären Einsätzen, dem Vermessungs- und Katasterwesen bis hin zum Umwelt- und Naturschutz, um nur einige wichtige Bereiche zu nennen. Auch für die Planung einer nachhaltigen Energieproduktion und Versorgung durch regenerative Energieformen, wie z.B. der Standortbestimmung von Windenergieanlagen (WEAs) und deren Anbindung an vorhandene bzw. neu zu errichtende Energienetze, werden mittlerweile vorwiegend GIS verwendet. Somit leisten GIS einen wesentlichen Beitrag bei den unterschiedlichsten Planungsvorhaben und Projekten, wobei bei all diesen Anwendungsgebieten als gemeinsamer Nenner immer räumliche Beziehungen von Objekten im Mittelpunkt stehen. Das Potential von geographischen Informationssystemen wird bis dato jedoch immer noch nicht im vollem Ausmaß genutzt, wobei sich die GIS-Softwareprodukte selbst rasant weiterentwickeln (LANG & BLASCHKE 2007). Die notwendige Hardware in Form von rechnergestützten Systemen entwickelt sich dabei, parallel zu den ständig wachsenden Anwendungsfeldern, ebenfalls rasant weiter und trägt somit gleichermaßen zu diesem enormen Entwicklungsfortschritt bei. Gleichzeitig hierzu nimmt auch der Funktionsumfang der dementsprechenden Softwareprodukte rasant zu. Generell variiert der Funktionsumfang zwischen den heutigen GIS-Softwareprodukten zum Teil stark, wobei dieser anhand der jeweils implementierten Algorithmen gemessen werden kann.

Algorithmen werden grundsätzlich zur Lösung unterschiedlichster Probleme bzw. Aufgabenstellungen benötigt und bilden sozusagen das zentrale Thema der Informatik. Die wichtigsten Eigenschaften von Algorithmen sind dabei einerseits deren Korrektheit, also dass sie die jeweils gestellte Aufgabe richtig lösen können, und andererseits deren Effizienz. Dazu gehören neben dem benötigten Speicherplatz auch die für den Berechnungsprozess benötigte Rechenzeit (OTTMANN & WIDMAYER 2017).

In Hinblick auf geographische Informationssysteme wird des Öfteren der Begriff (Geo-)Algorithmen verwendet, da mit ihrer Hilfe vorrangig geographische Aufgabenstellungen gelöst werden. Diese Algorithmen bilden die Grundbausteine jeder GIS-Applikation und werden u.a. in Form von Werkzeugen, sogenannten „*Geoprocessing-Tools*“, in diese Softwareprodukte implementiert. Die programmspezifische Gliederung der (Geo-)Algorithmen erfolgt dabei meist auf Basis der unterschiedlichen Aufgabengebiete.

Zu den Kernaufgabengebieten zählen Vektor- bzw. Rasterbearbeitungsalgorithmen, Algorithmen zur geostatistischen Analyse von räumlichen Datensätzen, Konvertierungsalgorithmen, Datenmanagementalgorithmen, Netzwerkanalysealgorithmen sowie die große Gruppe der Interpolationsalgorithmen welche zur Modellierung von digitalen Oberflächen verwendet werden.

Interpolationsalgorithmen werden grundsätzlich zur Erstellung von digitalen Gelände- bzw. Oberflächenmodellen (DGM) benötigt. Diese beschreiben jeweils Ausschnitte der realen Welt und sind zum Aufbau einer umfassenden Struktur von Geodaten notwendig (BARTELME 2005). Hierfür werden grundsätzlich Vektordatensätze, in Form von Punkt(mess)daten als Eingangsdatenquelle, verwendet und anschließend mit Hilfe eines Interpolationsverfahrens in Rasterdatensätze umgewandelt, wobei die Zellenwerte der damit generierten Raster schrittweise interpoliert werden.

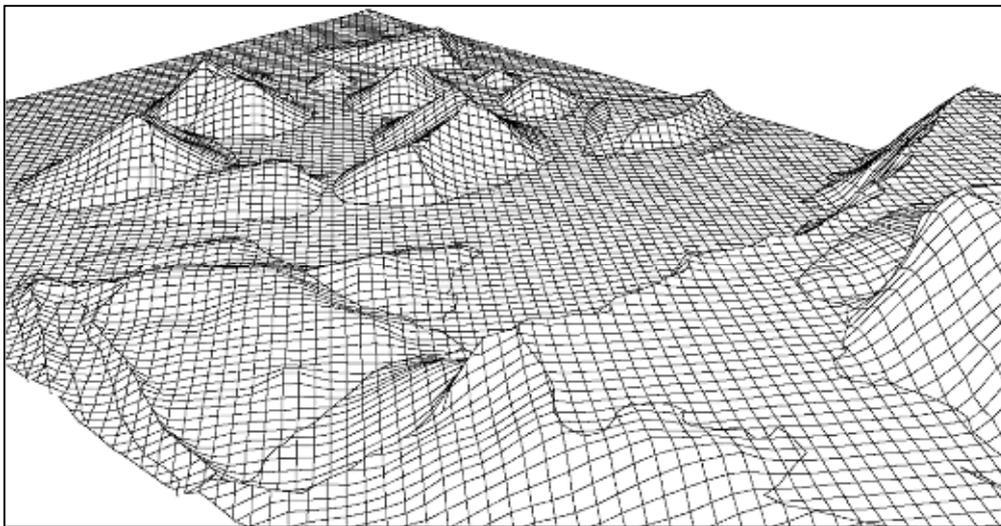


Abbildung 1: Beispiel für ein digitales Geländemodell (DGM)  
Quelle: BARTELME (2005)

Neben der Erstellung von digitalen Geländemodellen werden Interpolationsalgorithmen noch für weitere Anwendungsfälle, wie der Erzeugung unterschiedlicher Arten von Konzentrationsoberflächen (Schwermetallbelastung in Böden, Ozonbelastung etc.) oder auch für die Darstellung von Niederschlagsoberflächen, eingesetzt. Die Modellierung der Oberfläche erfolgt dabei auf Basis einer, für den jeweiligen Anwendungsfall, geeigneten Interpolationsmethode, wobei hierfür unterschiedliche Verfahren bzw. Ansätze zur Verfügung stehen. Zu den am häufigst verwendeten Interpolationsverfahren zählen dabei die „*Inverse Distance Weighted*“-Interpolation („*IDW*“), die „*Kriging*“-Interpolation, eine Interpolation anhand der „*Natural Neighbor*“-Methode („*NN*“), die „*Radial Basis Funktion*“ („*RBF*“), die „*Spline*“-Interpolation sowie die Erstellung einer „*Triangular Irregular Network*“-Oberfläche („*TIN*“). Diese und weitere Interpolationsalgorithmen sind standardmäßig in allen GIS-Softwareprodukten, in Form von eigenen „*Geoprocessing-Tools*“, implementiert und gehören somit zu den Basisalgorithmen jeder heutzutage gängigen GIS-Applikation. Weiters stehen meist mehrere verschiedene Varianten pro Verfahren zur Verfügung, wobei die Wahl der Interpolationsmethode entscheidenden Einfluss auf die Resultate hat.

## 1.2 Literaturüberblick

Vergleiche und Gegenüberstellungen der zuvor beschriebenen Interpolationsverfahren und deren zugrunde liegenden (Geo-)Algorithmen finden sich in zahlreichen wissenschaftlichen Publikationen und Abhandlungen.

Bereits vor mittlerweile 20 Jahren beschäftigten sich ZIMMERMAN et al. (1999) mit dem Thema der Interpolationsalgorithmen und den potentiell unterschiedlichen Resultaten bei deren Anwendung. Sie verglichen dabei den „*Inverse Distance Weighted*“-Algorithmus mit dem „*Kriging*“-Algorithmus, wobei jeweils zwei unterschiedliche Varianten dieser Interpolationsverfahren getestet wurden. Beim „*Kriging*“-Verfahren wurden die beiden grundsätzlichen Typen „*Ordinary*“- bzw. „*Universal-Kriging*“ verwendet und beim „*IDW*“-Verfahren einerseits die nächstgelegenen 6 Beobachtungspunkte und andererseits die nächstgelegenen 12 Punkte. Somit beinhaltet die Analyse den Vergleich von zwei unterschiedlichen Interpolationsalgorithmen („*IDW*“ bzw. „*Kriging*“), wobei hierfür experimentale „synthetische“ Daten für die Gegenüberstellung der Interpolationsverfahren verwendet wurden. Einerseits kamen drei mathematische Flächen („*Plane*“, „*Sombbrero*“ bzw. „*Morrison`s Surface*“) und andererseits jeweils vier verschiedene Arten von Punktemustern (u.a. ein regelmäßiges hexagonales Gitter, ein zufälliges und ein gruppiertes Muster) zum Einsatz. Um die Effizienz und Genauigkeit der einzelnen Interpolationsmethoden beurteilen zu können wurde bei allen erstellten Interpolationsvarianten jeweils der natürliche Logarithmus des „*Mean Square Error*“ („*log[MSE]*“), der „*Root Mean Square Error*“ („*RMSE*“) sowie der „*Mean Absolute Error*“ („*MAE*“) berechnet und gegenübergestellt. Das Ergebnis war, dass beide „*Kriging*“-Varianten („*Ordinary*“ und „*Universal*“) wesentlich bessere Resultate lieferten als die beiden „*IDW*“-Varianten. Dies ist bereits in den resultierenden Fehlerwerten deutlich erkennbar.

Um die Verteilung von Sulfat-Konzentrationen im oberflächennahen Grundwasser darzustellen verwendeten AZOUZI & SCHRÖTER (2012) die drei Interpolationsmethoden „*Ordinary Kriging*“, „*Fuzzy Kriging*“ und „*Natural Neighbor*“ und verglichen die damit erstellten Interpolationsoberflächen anschließend mit einer konventionell erstellten Karte. Für den Methodenvergleich wurde als Untersuchungsgebiet Bremen-Nord nordwestlich der Weser ausgewählt, wobei hierfür zwei Basisdatensätze (Variante 1 bzw. 2) mit jeweils unterschiedlicher Punktedichte verwendet wurden. Neben der Berechnung der Interpolationsoberflächen wurde auch eine Sensitivitätsanalyse durchgeführt um die optimalen Parameter für die Regionalisierung festzulegen. Für die Berechnungen und Analysen wurden dabei „*Variowin*“, „*SURFER 8*“ bzw. „*FUZZEKS*“ verwendet. Die Schlussfolgerung dieser wissenschaftlichen Untersuchung war, dass grundsätzlich alle hierfür verwendeten Interpolationsverfahren zur Berechnung und Darstellung der Sulfatgehalte geeignet sind, wobei bei der „*Natural Neighbor*“-Methode die Abhängigkeit von der Datendichte am größten ist.

Das „*Fuzzy Kriging*“-Verfahren verbindet andererseits die Qualität eines geostatistischen Interpolationsverfahrens mit der Möglichkeit unscharfe Daten zu verarbeiten und liefert daher letztendlich, in Hinblick auf den untersuchten Sachverhalt, die besten Interpolationsresultate.

ARUN (2013) beschäftigte sich in seiner Arbeit mit dem Vergleich von insgesamt fünf der am häufigst verwendeten Interpolationsmethoden zur Erstellung von digitalen Oberflächenmodellen. Neben der „*Inverse Distance Weighted*“-Interpolation, „*ANUDEM*“, „*Nearest Neighbor*“ und „*Spline*“ wurde weiters auch „*Ordinary-Kriging*“ in Hinblick auf dessen Interpolationsergebnisse untersucht. Hierfür wurden mehrere Basisdatensätze (u.a. „*Survey of India*“ („*SOI*“) bzw. „*SRTM*“-Daten) verwendet, Referenzpunkte definiert und deren Höhen mit Hilfe der fünf unterschiedlichen Interpolationsalgorithmen berechnet, und anschließend der daraus resultierende „*Root Mean Square Error*“ je Methode und Datensatz ermittelt. Die Gegenüberstellung zeigt, dass, bezogen auf die verwendeten Datensätze, jeweils der „*Kriging*“-Interpolationsalgorithmus den geringsten „*RMSE*“ generiert und somit bei diesem Vergleich die besten Ergebnisse liefert. Weiters wird auch noch darauf hingewiesen, dass die hierfür verwendeten Datensätze und deren grundlegenden Charakteristika einen wesentlichen Einfluss auf das Ergebnis dieser Untersuchung haben.

ADHIKARY & DASH (2017) untersuchten anhand der „*IDW*“-Interpolation, der „*Radial Basis Function*“ und „*Ordinary*“, bzw. „*Universal-Kriging*“ das Erstellen von Interpolationsoberflächen in Hinblick auf den Grundwasserspiegel. Als Untersuchungsgebiet diente dabei das „*National Capital Territory*“ von Delhi in Indien. Ähnliche Untersuchungen aus den Jahren 2003, 2007 und 2009 hatten gezeigt, dass für diese spezielle Aufgabenstellung nahezu ausnahmslos das „*Universal Kriging*“-Verfahren eingesetzt wurde. Um diese Vorgehensweise zu validieren, und um ggf. auf ein besser geeignetes Interpolationsverfahren in Hinblick auf die Erstellung von Grundwasserspiegel-Karten hinzuweisen, wurde diese Untersuchung durchgeführt. Die Kreuzvalidierung der Ergebnisse erfolgte auf Basis des „*Root Mean Square Errors*“, dem „*Mean Error*“ und dem „*Mean Relative Error*“. Die Gegenüberstellung der Fehlerwerte zeigt, dass „*Universal-Kriging*“ hier die niedrigsten Werte lieferte und somit bei diesem Vergleich wiederum am besten abschnitt.

Eine weitere wissenschaftliche Untersuchung von PAVLOVA (2017) beschäftigt sich mit dem Vergleich und der Analyse von insgesamt sechs verschiedenen Interpolationsverfahren: der „*Inverse Distance Weighted*“-Interpolation, „*Kriging*“ („*Ordinary*“, bzw. „*Universal*“), „*ANUDEM*“, der „*Spline*“-Interpolation („*Regularized*“ bzw. „*Tension*“) und der „*Natural Neighbor*“-Methode. Weiters wird auch noch das „*Triangular Irregular Network*“-Modell untersucht. Für die Berechnungen und der anschließenden Analyse der unterschiedlichen Interpolationsoberflächen wurde *ES-RIs ArcGIS 10* inkl. der *Extensions* „*Spatial Analyst*“ und „*3D Analyst*“ verwendet. Als Datengrundlage wurden unterschiedliche Basisdatensätze von der „*Omsk*“-Region in West Sibirien herangezogen, wobei hierfür eine Untersuchungsfläche von insgesamt 8742 km<sup>2</sup> verwendet wurde.

Um die Resultate der Untersuchung zu validieren wurden je Basisdatensatz Referenzpunkte entnommen und anschließend jeweils der „Mean Error“ („ME“), der „Mean Absolute Error“ („MAE“) und der „Root Mean Square Error“ („RMSE“) für jede Interpolationsoberfläche berechnet und tabellarisch gegenübergestellt. Die Kreuzvalidierung zeigt, dass jeweils die beiden „Spline“-Varianten „Regularized“- bzw. „Tension“-Spline, gefolgt vom „IDW“-Verfahren die geringsten Fehlerwerte generierten und somit die besten Interpolationsergebnisse lieferten.

QIAO et al. (2018) nutzten das „IDW“- und das „Ordinary Kriging“-Verfahren für die Erstellung von räumlichen Karten bezüglich der Bodenverschmutzung von Peking in China, wobei als Datengrundlage statistische Erhebungen aus dem Jahr 2000 verwendet wurden. Die anhand der resultierenden Interpolationsoberflächen ermittelten Fehlerwerte („MAE“, „MRE“ bzw. „RMSE“) unterscheiden sich dabei je Interpolationsvariante nur minimal voneinander. Aufgrund der sehr geringen Anzahl von lediglich 146 Eingangsdatenpunkten (Messpunkten vor Ort) ist lt. dieser Untersuchung letztlich das „IDW“-Interpolationsverfahren am besten für die räumliche Vorhersage der Schwermetallbelastung in den Böden von Peking geeignet.

Um eine regionalisierte Darstellung der Nitratbelastung im Grundwasser Niedersachsens auf Basis von Punktmessdaten zu erstellen, verwendeten WRIEDT et al. (2019) das „Ordinary Kriging“- und das „Simple Kriging“-Verfahren, wobei jeweils verschiedene Semivariogramm-Modelle getestet wurden. Letztlich wird auf die deutlich unterschiedlichen Interpolationsergebnisse, und auf eine zukünftige Forschung hinsichtlich der Weiterentwicklung der verwendeten Modelle, hingewiesen.

Anhand des Literaturüberblicks ist ersichtlich, dass Vergleiche unterschiedlicher Interpolationsverfahren bereits seit vielen Jahren durchgeführt werden, jedoch wieder erwarten diese meist zu unterschiedlichen Aussagen bzw. Resultaten führten. Daher ist es aus wissenschaftlicher Sicht durchaus gerechtfertigt sich anhand der vorliegenden Master Thesis wiederum mit dieser Thematik zu befassen um dadurch möglichst zu einem besseren Verständnis hinsichtlich der verschiedenen Interpolationsalgorithmen und deren Anwendung beizutragen.

### 1.3 Zielsetzung und Forschungsfrage

In Hinblick auf das zuvor beschriebene Themenfeld, dessen Relevanz und der im Literaturüberblick dargestellten Thematik lauten die drei zu beantwortenden Forschungsfragen wie folgt:

- 1. Welche Verfahren zur Interpolation von digitalen Oberflächen gibt es?**
- 2. Ergeben sich bei der Gegenüberstellung der drei gängigsten Interpolationsalgorithmen Unterschiede in Bezug auf deren Interpolationsergebnisse?**
- 3. Welche dieser drei Interpolationsalgorithmen liefert hinsichtlich der Topographie und der Messpunktdichte die besten Ergebnisse?**

Das Ziel der vorliegenden Master Thesis ist somit eine Gegenüberstellung der drei gängigsten Interpolationsalgorithmen und deren potentiell unterschiedlichen Interpolationsergebnisse.

Folgende drei Interpolationsalgorithmen bzw. Verfahren werden ausgewählt und verglichen:

- „***INVERSE DISTANCE WEIGHTING***“ („***IDW***“)
- „***KRIGING***“
- „***SPLINE***“

Um die Forschungsfragen 2 und 3 beantworten zu können wird ein mehrstufiger automatisierter Geoprozess benötigt, welcher mit Hilfe einer repräsentativen GIS-Applikation und deren standardmäßig integrierten Entwicklungsumgebung erstellt wird. Das Python Add-In dient dabei einerseits der Berechnung aller Interpolationsoberflächen und andererseits zur Kreuzvalidierung der Interpolationsergebnisse. Weiters wird auch eine praxisbezogene Anwendung ermöglicht, wobei der Fokus der vorliegenden Arbeit jedoch primär auf den Vergleich der drei unterschiedlichen Interpolationsalgorithmen bzw. Verfahren gerichtet ist.

Der Geoprozess wird mit insgesamt drei unterschiedlichen Basisdatensätzen als Eingangsdaten(quellen) ausgeführt und anschließend die hiermit erzeugten Interpolationsoberflächen gegenübergestellt und analysiert. Im Sinne einer Sensitivitätsanalyse wird dabei untersucht, ob die jeweils ermittelten Ergebnisse und die daraus resultierenden Erkenntnisse aufgrund der unterschiedlichen Eingangsdaten(quellen) variieren und diese somit einen entscheidenden Einfluss auf die Resultate haben. Weiters wird anhand der Sensitivitätsanalyse die Empfindlichkeit der einzelnen Interpolationsalgorithmen selbst übergeprüft, wobei dies mit Hilfe einer unterschiedlichen Gewichtung deren (Haupt-)Eingabeparameter anhand von vordefinierten Parameterräumen überprüft wird.

Zur Beantwortung der Forschungsfragen sind folgende operativen Teilschritte notwendig:

1. Die Wahl einer repräsentativen GIS-Applikation zur Umsetzung des anvisierten Geoprozesses und somit die, zum Erstellen und Analysieren der einzelnen Interpolationsoberflächen, verwendete GIS-Software.
2. Die Beschreibung der theoretischen Grundlagen der einzelnen Interpolationsverfahren.
3. Die Definition und Spezifikation des Geoprozesses und der, für den Vergleich der drei Interpolationsalgorithmen und deren Resultate, benötigten Basisdatensätze.
4. Die Analyse und Beurteilung der Interpolationsergebnisse anhand
  - a. einer visuellen Analyse der Ergebnisse,
  - b. einer Explorativen Räumlichen Datenanalyse („*ESDA*“),
  - c. und einer Kreuzvalidierung aller Interpolationsergebnisse.

## 1.4 Abgrenzung

Im Rahmen der vorliegenden Arbeit muss eine Abgrenzung hinsichtlich der im weiteren Verlauf verwendeten GIS-Software vorgenommen werden. Da in der Industrie und Praxis zur Lösung von komplexen Aufgabenstellungen mit geographischem Kontext nach wie vor vorrangig Desktop-GIS Applikationen eingesetzt werden liegt der Fokus der vorliegenden Master Thesis auf dieser Art von Software-Produkten. Alle Berechnungen, Untersuchungen und Analysen werden standardmäßig mit Hilfe von ESRI's ArcGIS Pro durchgeführt. Dieses Software-Produkt wird von ESRI als die führende professionelle Desktop-GIS-Anwendung in deren Produkt-Portfolio vertrieben. Es wird hierfür ständig weiterentwickelt und entspricht somit dem momentanen Stand der Technik. In der vorliegenden Arbeit wird dabei nur auf jene Bereiche von ArcGIS Pro eingegangen, welche zur Lösung der Aufgabenstellung und somit zur Beantwortung der Forschungsfragen notwendig sind.

Im Zuge dieser Master Thesis wird folgende GIS-Applikation verwendet:

- **ArcGIS Pro – Version 2.2.4 Advanced 64 Bit** – mit folgenden Extensions:
  - „3D Analyst“ – Version 2.2.4.12813
  - „Data Interoperability“ – Version 2.2.4.12813
  - „Data Reviewer“ – Version 2.2.4.12813
  - „Geostatistical Analyst“ – Version 2.2.4.12813
  - „Network Analyst“ – Version 2.2.4.12813
  - „Spatial Analyst“ – Version 2.2.4.12813
  - „Workflow Manager“ – Version 2.2.4.12813

In älteren wissenschaftlichen Untersuchungen wird zum Teil auf die unterschiedlichen Ergebnisse bei der Verwendung gleicher Werkzeuge, und deren zugrunde liegenden (Geo-)Algorithmen, in verschiedenen Softwareprodukten hingewiesen. Des Weiteren sind diese Algorithmen für den Anwender der Softwareprodukte meist nicht oder nur sehr schwer einsehbar bzw. in der jeweiligen Dokumentation nachlesbar. FISHER (1993) beschäftigte sich mit dieser Thematik bereits in den 90er Jahren des letzten Jahrhunderts und weist dabei auf die zum Teil gravierenden Unterschiede bei den Ergebnissen in Bezug auf die Implementierung und Anwendung potentiell gleicher (Geo-)Algorithmen in unterschiedlichen GIS-Softwareprodukten hin. Er untersuchte in diesem Zusammenhang anhand von zwei unterschiedlichen Testgebieten bzw. Datensätzen speziell die Implementierung des „Viewshed“-Algorithmus in insgesamt sieben verschiedenen GIS-Applikationen, wobei jeweils drei Hauptmerkmale analysiert wurden. Dabei wurde teilweise eine erhebliche Variabilität in Bezug auf den sichtbaren Bereich und der Ergebnisse der verwendeten Algorithmen beobachtet.

Eine aus diesem Grund vom Verfasser der vorliegenden Arbeit durchgeführte Metastudie, welche das Ziel hatte unterschiedliche Realisierungen gängiger (Geo-)Algorithmen (u.a. den „*Inverse Distance Weighted*“-Interpolationsalgorithmus) in zwei verschiedenen Desktop-GIS Applikationen (ArcGIS Pro bzw. QGIS) zu untersuchen, kam zum Schluss, dass letztlich nur geringfügigste Abweichungen zwischen den Endergebnissen festzustellen waren. Alle im Verlauf der Studie durchgeführten Vergleiche zeigten, dass die Implementierung und Realisierung der insgesamt vier untersuchten (Geo-)Algorithmen in den beiden Applikationen nahezu idente Resultate lieferten!

Die Ergebnisse der vorliegenden Arbeit sind somit grundsätzlich unabhängig von der verwendeten GIS-Software zu betrachten. Bei Verwendung einer anderen gleichwertigen Desktop-GIS Applikation, welche ebenfalls den heutigen Stand der Technik entspricht, ist demnach, im Sinne der Allgemeingültigkeit der Resultate nicht zu erwarten, dass sich relevante Unterschiede ergeben würden.

Bei der nachfolgenden Ausarbeitung der Master Thesis ist unter Umständen noch eine weitere (thematische) Abgrenzung notwendig, wobei diese dann im jeweiligen Kapitel eindeutig gekennzeichnet und festgehalten wird.

## 1.5 Aufbau und Struktur der Arbeit

Die vorliegende Arbeit gliedert sich in insgesamt sieben Kapitel.

Das erste Kapitel stellt die Einleitung dar. Es wird zuerst auf das Themenfeld und dessen Relevanz eingegangen. Anschließend folgt ein Literaturüberblick um die Ergebnisse bereits durchgeführter wissenschaftlicher Untersuchungen in Hinblick auf den Vergleich von Interpolationsverfahren aufzuzeigen. Weiters werden die Zielsetzung bzw. die Forschungsfragen und der methodische Ansatz zur Lösung spezifiziert. Eine Abgrenzung der Arbeit ist ebenfalls Bestandteil dieses Kapitels.

Nach der Einleitung wird im zweiten Kapitel auf die theoretischen Grundlagen eingegangen. Hierfür werden die grundlegenden Spezifika der verwendeten Interpolationsalgorithmen erläutert und auf die Vor- bzw. Nachteile der jeweiligen Verfahren hingewiesen.

Kapitel 3 befasst sich mit den, für den Vergleich der Interpolationsverfahren, notwendigen Basisdatensätzen. Neben allgemeinen Informationen hinsichtlich der drei verwendeten Testdatensätze werden deren spezifischen Charakteristika (Topographie, Datendichte etc.) beschrieben.

Kapitel 4 beschäftigt sich mit der Struktur des Geoprozesses, welcher zum Erstellen der Interpolationsoberflächen und für die Kreuzvalidierung der Ergebnisse verwendet wird. Weiters wird auch noch auf die durchzuführende Sensitivitätsanalyse in diesem Kapitel näher eingegangen.

Kapitel 5 dokumentiert die Interpolationsergebnisse, wobei diese getrennt nach Basisdatensatz dargestellt werden. Neben der Darstellung der berechneten Interpolationsoberflächen, einer explorativen räumlichen Datenanalyse aller Resultate, umfasst dieses Kapitel auch die Kreuzvalidierung der Ergebnisse.

Kapitel 6 befasst sich mit der Diskussion der Interpolationsergebnisse, der Beantwortung der gestellten Forschungsfragen und dem Vergleich der anhand dieser Master Thesis gewonnenen Erkenntnisse mit den Ergebnissen der bisher durchgeführten Studien bzw. Untersuchungen.

Das abschließende Kapitel 7 fasst die wichtigsten Forschungsergebnisse zusammen und gibt einen Ausblick über mögliche zukünftige Entwicklungen und Forschungstätigkeiten.

In Hinblick auf eine lückenlose Reproduzierbarkeit der Ergebnisse der vorliegenden Master Thesis befindet sich im Anhang ein vollständiger Ausdruck des ArcGIS Pro Python Add-Ins. Des Weiteren steht der für den Vergleich der Interpolationsalgorithmen entworfene automatisierte Geoprozess, alle in der vorliegenden Arbeit verwendeten Testdatensätze (siehe Kapitel 3) und die zum Ausführen des Geoprozesses notwendige Ordnerstruktur unter folgenden Link zum Download bereit:

<https://www.dropbox.com/sh/4716lur06zgwahaa/AAAF17s6oJVMwlTyGYuoqxIsa?dl=0>

## 2. Theoretische Grundlagen

### 2.1 Grundlagen und historische Entwicklung von Interpolationsalgorithmen

Für die Interpolation von Oberflächen werden grundsätzlich die Zellenwerte des zu erstellenden (Ziel-)Rasters auf Basis einer begrenzten Menge von bekannten (Mess-)Punkten ermittelt, wobei hierfür prinzipiell unterschiedliche Verfahren bzw. Interpolationsalgorithmen zur Verfügung stehen. Die mit Hilfe dieser Methoden erstellten Interpolationsoberflächen sollen eine bestmögliche Annäherung an die von uns wahrgenommene Wirklichkeit gewährleisten.

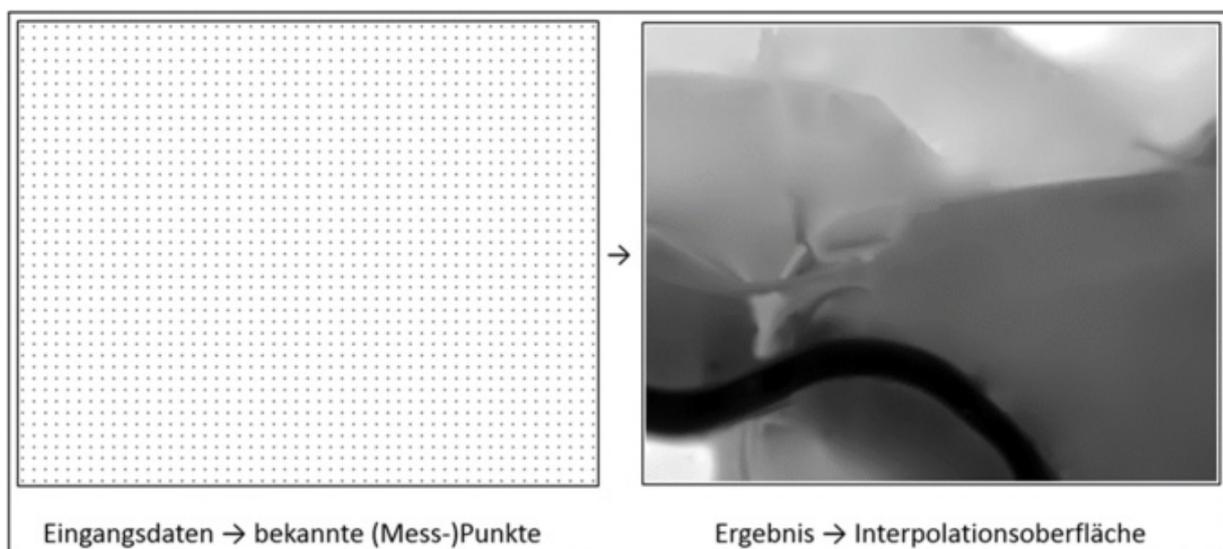


Abbildung 2: Interpolation einer Höhenoberfläche auf Basis von bekannten (Mess-)Punkten

Grundsätzlich können alle Interpolationsverfahren und deren zugrunde liegenden (Geo-)Algorithmen in zwei große Gruppen aufgeteilt werden. Die erste Gruppe bilden dabei die sogenannten **deterministischen Interpolationsverfahren**. Zu diesen zählen u.a. die „*Inverse Distance Weighted Interpolation*“ („*IDW*“), die „*Global Polynomial Interpolation*“ („*GPI*“), die „*Local Polynomial Interpolation*“ („*LPI*“), „*Natural Neighbor*“ („*NN*“), die „*Radial Basis Funktion*“ („*RBF*“) sowie die „*Spline*“-Interpolation. All diese Verfahren haben gemeinsam, dass zur Ermittlung der Zellenwerte (Schätzwerte) des (Ziel-)Rasters lediglich die Distanz zu den (Mess-)Punkten berücksichtigt wird, wobei die räumliche Nähe unterschiedlich gewichtet wird (EISANK & KOCH 2017a).

Bei den deterministischen Interpolationsverfahren werden des Weiteren keine potentiell möglichen räumlichen Autokorrelationen zwischen den (Mess-)Punkten berücksichtigt und die Berechnungen werden somit rein auf Basis von mathematischen Funktionen durchgeführt.

Die zweite Gruppe von Interpolationsmethoden bilden die sogenannten **probabilistischen Interpolationsverfahren**. Hierbei handelt es sich um eine Kombination mathematischer und statistischer Verfahren. Man spricht daher auch von geostatistischen Interpolationsmethoden. Die Ermittlung der Zellenwerte (Schätzwerte) des (Ziel-)Rasters beruht, anders als bei den deterministischen Verfahren, auf Wahrscheinlichkeiten. Es werden neben den Distanzen zwischen den (Mess-)Punkten und den zu ermittelnden Zellenwerten (Schätzwerten) auch die räumlichen Autokorrelationen der (Mess-)Punkte untereinander mitberücksichtigt (DAYA & BEJARI 2015).

Somit hat die räumliche Konfiguration der (Mess-)Punkte einen entscheidenden Einfluss auf das Ergebnis dieser Interpolationsalgorithmen. Das bekannteste und zugleich am häufigst verwendete Verfahren dieser zweiten Gruppe ist das „*Kriging*“-Interpolationsverfahren.

Folgende „*Kriging*“-Verfahren stehen, je nach (vermuteten) räumlichem Trend im Untersuchungsgebiet, für die Berechnung der resultierenden Interpolationsoberflächen zur Verfügung:

- „*Ordinary Kriging*“
- „*Simple Kriging*“
- „*Universal Kriging*“
- „*Indicator Kriging*“
- „*Co-Kriging*“

Neben den Interpolationsoberflächen als Ergebnis werden bei Anwendung des „*Kriging*“-Verfahrens auch noch die dazugehörigen Fehlerflächen (die sogenannten „*Error-Surfaces*“) erstellt, womit eine anschließende Beurteilung der ermittelten Zellenwerte ermöglicht wird.

Zusammenfassend kann festgehalten werden, dass, in Hinblick auf die beiden Gruppen von Interpolationsverfahren und deren jeweils angehörigen Interpolationsalgorithmen, es eine Vielzahl unterschiedlicher Methoden zur Erstellung von Interpolationsoberflächen gibt und erst durch den Vergleich der einzelnen Verfahren eine Hilfestellung für potentielle Anwender gegeben werden kann.

Historisch betrachtet sind die meisten der heute in den GIS-Applikationen implementierten Interpolationsalgorithmen bzw. Interpolationsverfahren nicht neu, sondern Großteils bereits seit Jahrzehnten bekannt. So wurde der grundlegende Ansatz zum heutigen „*IDW*“-Interpolationsverfahren bereits vor mehr als 50 Jahren von SHEPARD (1968) entwickelt und auch publiziert. Daher wird diese Interpolationsmethode auch als Shepard-Interpolation bezeichnet. Ähnliches gilt auch für den probabilistischen „*Kriging*“-Interpolationsalgorithmus. Diese Interpolationsmethode wurde bereits im Jahr 1966 von Herrn Danie G. Krige, einem südafrikanischen Bergbauingenieur, erstmals öffentlich beschrieben und wurde später auch nach ihm benannt (KRIGE 1966).

## 2.2 Interpolationsalgorithmen

### 2.2.1 „INVERSE DISTANCE WEIGHTED“ („IDW“)

Bei diesem (Geo-)Algorithmus handelt es sich um eines der am häufigst verwendeten deterministischen Interpolationsverfahren, wobei diese Interpolationsmethode in allen heutzutage gängigen GIS-Applikationen implementiert ist. Mit Hilfe dieses Interpolationsverfahrens werden die zu ermittelten Zellenwerte anhand einer linearen Gewichtung benachbarter Referenz- bzw. Messpunkte berechnet, wobei die Gewichtung abhängig von der inversen Distanz vorgenommen wird (BURROUGH & MCDONNELL 1998). Je größer die Distanz zwischen dem zu schätzenden Punkt und dem jeweiligen Referenz- bzw. Messpunkt ausfällt, desto geringer ist dessen Einfluss, wobei mit Hilfe des „Potenz“-Parameters die Gewichtung dieses Abstandes zusätzlich beeinflusst werden kann (ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE 2018b).

Einen wesentlichen Einfluss auf das Ergebnis der Interpolation hat weiters der verwendete Suchradius. Dieser kann entweder variabel oder aber als fester Suchradius definiert werden. Somit wird entweder eine fixe Anzahl von Punkten, welche für die Berechnung der zu interpolierenden Zellen verwendet wird, spezifiziert oder ein fixer Radius. In diesem Fall werden alle innerhalb des definierten Suchradius befindlichen Punkte zur Ermittlung der jeweiligen Zellenwerte herangezogen.

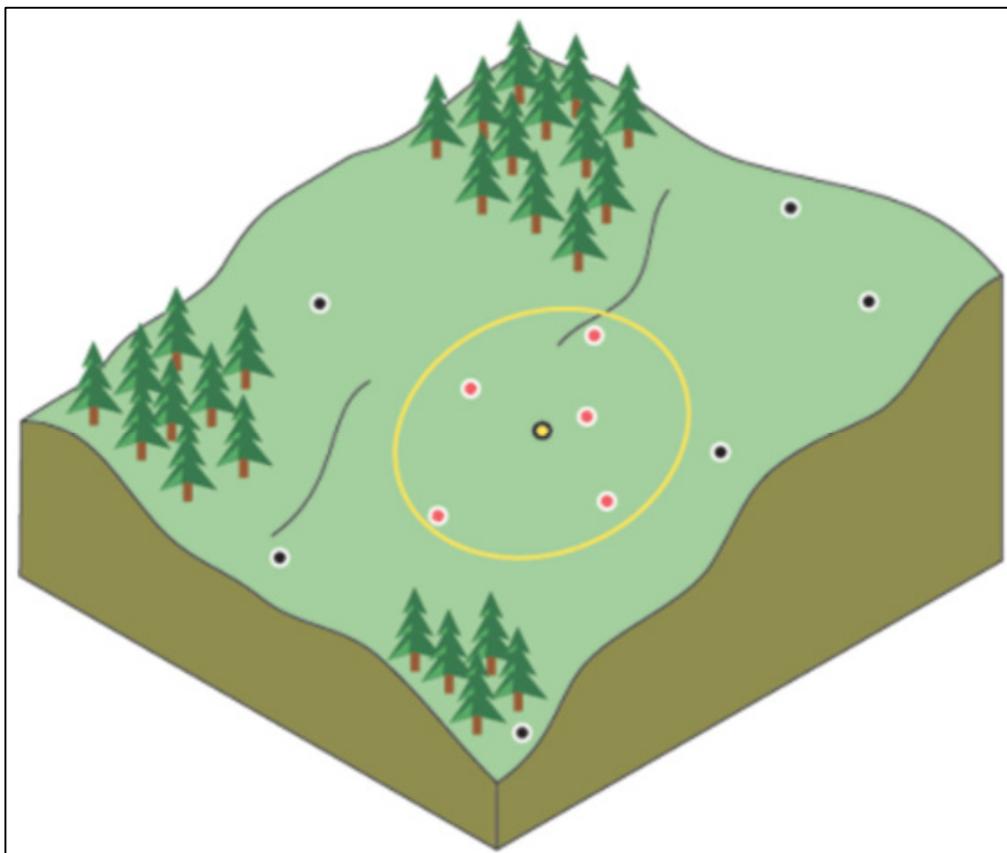


Abbildung 3: IDW-Nachbarschaft für einen definierten Punkt  
Quelle: ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2018b)

Beim „IDW“-Interpolationsalgorithmus erfolgt grundsätzlich die Berechnung des jeweiligen **Schätz- bzw. Zellenwertes**  $v$  bezogen auf den jeweiligen **Punkt P** anhand folgender Berechnungsformel:

$$\hat{v}_P = \frac{\sum_{i=1}^n \frac{1}{d_i^p} v_{Q_i}}{\sum_{i=1}^n \frac{1}{d_i^p}}$$

Quelle: EISANK & KOCH (2017a)

wobei:

- $n$  → die Anzahl der Punkte im Suchradius,
- $d_i^p$  → die Distanz zwischen den Punktepaaren,
- $v_{Q_i}$  → „*Sample Point*“ innerhalb des Suchradius,
- $p$  → den „Potenz“-Parameter darstellt.

Aus obiger Berechnungsformel ist dabei ersichtlich, dass die jeweiligen Distanzen umgekehrt proportional gewichtet werden. Mit Hilfe des sogenannten Gewichtungsparameters der Distanz („Potenz“- bzw. „*Power*“-Parameter  $p$ ) kann des Weiteren die Distanz potenziert werden. Je größer der Wert dieses Parameters gewählt wird, desto stärker nehmen die Gewichte mit zunehmender Distanz ab und somit desto zerklüfteter wird die jeweilige Schätz- bzw. Interpolationsoberfläche (EISANK & KOCH 2017a).

Ein wesentlicher Nachteil dieses Interpolationsverfahrens ist, dass keine (möglichen) räumlichen Autokorrelationen zwischen den zu ermittelnden Punkten bzw. Zellen berücksichtigt werden, wie es z.B. beim probabilistischen „*Kriging*“-Interpolationsalgorithmus der Fall ist, und der Gewichtungsparameter der Distanz („*Power*“-Parameter) einheitlich über die gesamte zu interpolierende Oberfläche definiert wird. Es gibt jedoch wissenschaftliche Studien die sich mit variierenden Gewichtungsparametern beschäftigen, um zu versuchen den Einfluss der unterschiedlichen Verteilung der Referenz- bzw. Messpunkte besser berücksichtigen zu können.

LU & WONG (2008) beschreiben in ihrem Artikel eine Methode zur anpassungsfähigen „IDW“-Interpolation („*Adaptive Inverse Distance Weighting*“ kurz „*AIDW*“), wobei sie hierfür einen eigenen Algorithmus entwickelt haben. Mit Hilfe dieses Algorithmus werden die optimalen anpassungsfähigen Gewichtungparameter der Distanz ermittelt, wobei das Grundprinzip der „IDW“-Interpolation weiterhin bestehen bleibt. Anhand von zwei unterschiedlichen Beispielen wird die praktische Anwendung der „*AIDW*“-Methode demonstriert und die Ergebnisse anschließend analysiert, wobei jeweils auch das Ergebnis des „*Kriging*“-Algorithmus dargestellt wird. Beim ersten Beispiel handelt es sich um Regenmessdaten um eine Niederschlagskarte von Taiwan zu erstellen. Das zweite Beispiel befasst sich mit der Erstellung einer Geländeoberfläche von Virginia County.

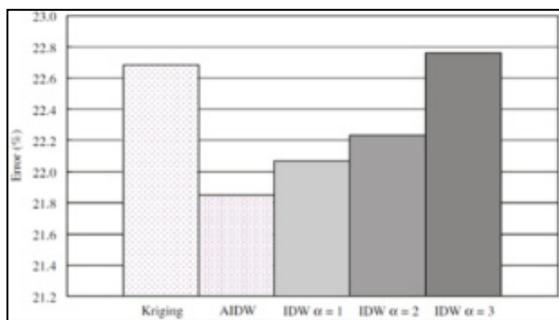


Abbildung 4: Beispiel 1 – resultierende Interpolationsfehler  
Quelle: LU & WONG (2008)

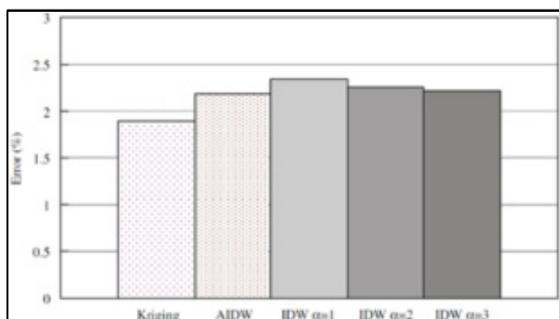


Abbildung 5: Beispiel 2 – resultierende Interpolationsfehler  
Quelle: LU & WONG (2008)

Anhand der Abbildungen 4 und 5 ist ersichtlich, dass sich bei der „*AIDW*“-Methode der jeweils resultierende Interpolationsfehler im Verhältnis zu den standardisierten Interpolationsverfahren „*IDW*“ bzw. „*Kriging*“ zum Teil erheblich reduziert, wobei der Unterschied beim zweiten Beispiel bereits wesentlich geringer ausfällt. Somit sind hinsichtlich der „*AIDW*“-Interpolationstechnik grundsätzlich weitere wissenschaftliche Studien ratsam um dieses Interpolationsverfahren wissenschaftlich validieren zu können. In der vorliegenden Arbeit wird daher der bereits in allen GIS-Applikationen standardmäßig implementierte „*IDW*“-Interpolationsalgorithmus verwendet.

Grundsätzlich zeigen mehrere unabhängig voneinander durchgeführten Untersuchungen, welche sich mit dem unmittelbaren Vergleich der am häufigst verwendeten Interpolationsmethoden (u.a. „IDW“) und deren zugrunde liegenden (Geo-)Algorithmen beschäftigen, dass die Ergebnisse der einzelnen Interpolationsverfahren im direkten Vergleich deutlich unterschiedlich zueinander ausfallen können. ARUN (2013) beschäftigte sich bereits 2013 in seinem wissenschaftlichen Artikel mit dem Vergleich der fünf häufigst verwendeten Interpolationsverfahren (u.a. „IDW“, „Kriging“ und „Spline“). Als Datengrundlage verwendete er Satellitenbilder von „Bhopal“ in Indien, sowie „Survey of India“ („SOI“) bzw. „SRTM“-Daten. Die nachfolgende Abbildung 6 zeigt die mit Hilfe dieser Daten erzeugten Interpolationsoberflächen, wobei bereits anhand einer rein visuellen Auswertung bzw. Analyse deutliche Unterschiede zwischen den einzelnen Ergebnissen ersichtlich sind.

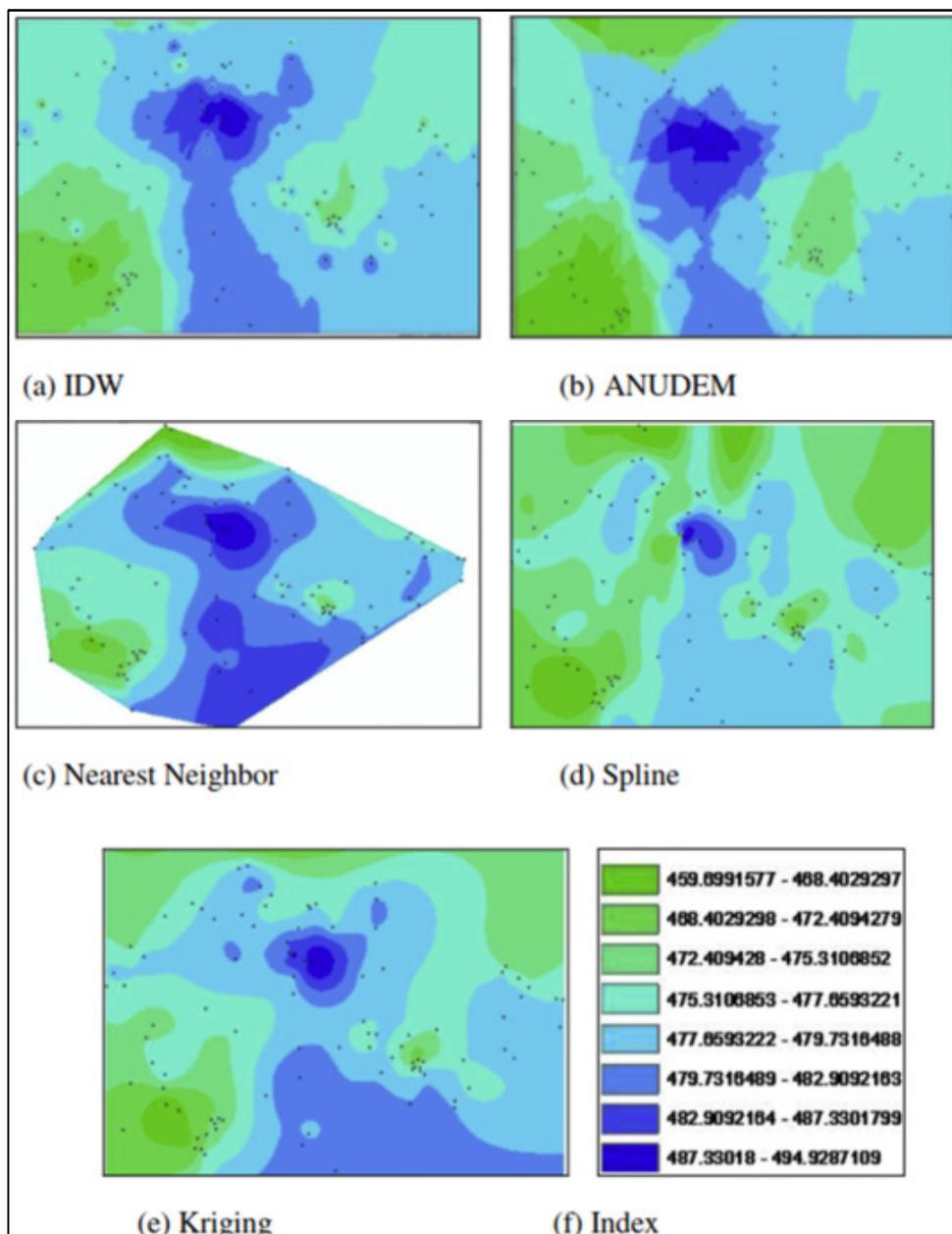


Abbildung 6: Interpolationsergebnisse  
Quelle: ARUN (2013)

### 2.2.2 „KRIGING“

„Kriging“ gehört zur Gruppe der probabilistischen Interpolationsverfahren und berücksichtigt grundsätzlich nicht nur die Distanzen zwischen dem Schätzpunkt und den Messpunkten, sondern zusätzlich auch die räumlichen Autokorrelationen der Messpunkte untereinander. Bei dieser Interpolationsmethode handelt es sich somit um eine Kombination mathematischer und statistischer Verfahren, wobei die Schätzwerte (Zellenwerte des Interpolationsrasters) auf Basis von Wahrscheinlichkeiten ermittelt werden. In diversen wissenschaftlichen Publikationen wird, im Zusammenhang mit diesem Interpolationsverfahren, oft der Ausdruck „B.L.U.E.“ für „*Best Linear Unbiased Estimator*“ verwendet, also die beste verzerrungsfreie Schätzmethode (AZOUZI & SCHRÖTER 2012).

Grundsätzlich arbeitet der „Kriging“-Algorithmus mit der Autokorrelation, also mit den geostatistischen Beziehungen zwischen den Messpunkten, wobei davon ausgegangen wird, dass die Entfernung bzw. Richtung zwischen den Referenzpunkten eine räumliche Korrelation darstellen. Dabei passt das Verfahren eine mathematische Funktion an eine gegebene Anzahl von Punkten, bzw. die Punkte innerhalb eines definierten Suchradius, an um daraus den Zellenwert für jede Position zu bestimmen (ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE 2016).

Um das „Kriging“-Interpolationsverfahren erfolgreich anzuwenden sind dabei mehrere Arbeitsschritte notwendig:

- eine statistische Analyse der Eingangsdaten,
- die Variogramm-Modellierung,
- die eigentliche Erstellung der Interpolationsoberfläche,
- die Kreuzvalidierung der Ergebnisse
- und ggf. die Untersuchung einer Varianzoberfläche.

Folgende „Kriging“-Varianten stehen dabei grundsätzlich zur Verfügung:

- „**Ordinary Kriging**“ („OK“),
- „**Simple Kriging**“ („SK“),
- „**Universal Kriging**“ („UK“),
- „**Indicator Kriging**“ („IK“),
- „**Co-Kriging**“,

wobei „Ordinary-“, bzw. „Universal-Kriging“ die am häufigst verwendeten Methoden dieser Gruppe von geostatistischen Interpolationsverfahren sind. Beide Varianten sind in allen gängigen GIS-Applikationen implementiert und werden somit auch in der vorliegenden Arbeit verwendet.

Das Grundprinzip vom „Kriging“-Interpolationsalgorithmus lautet wie folgt:

$$\mathbf{Z}(\mathbf{s}) = \boldsymbol{\mu}(\mathbf{s}) + \boldsymbol{\varepsilon}(\mathbf{s})$$

Quelle: EISANK & KOCH (2017b)

wobei:

- $\mathbf{s}$  → den räumlichen Ort repräsentiert,
- $Z(\mathbf{s})$  → die Variable  $Z$  am Ort  $\mathbf{s}$  ist,
- $\boldsymbol{\mu}(\mathbf{s})$  → der Mittelwert am Ort  $\mathbf{s}$  ist und
- $\boldsymbol{\varepsilon}(\mathbf{s})$  → den autokorrelierten Fehler am Ort  $\mathbf{s}$  darstellt.

Der Wert der Variable  $\mathbf{Z}$  am Ort  $\mathbf{s}$  ist somit durch den Mittelwert  $\boldsymbol{\mu}$  und einen räumlich autokorrelierten Zufallsfehler  $\boldsymbol{\varepsilon}$  an diesem Ort gegeben. Da der Mittelwert  $\boldsymbol{\mu}$  meist aufgrund eines auftretenden räumlichen Trends im Untersuchungsgebiet unbekannt ist wird dieser je nach „Kriging“-Variante auch unterschiedlich ermittelt.

Daher gilt für:

- „*Ordinary-Kriging*“ →  $\boldsymbol{\mu}(\mathbf{s}) = m$   
 $m \dots$  ist dabei unbekannt und hypothetisch konstant über das gesamte Untersuchungsgebiet.
- „*Simple-Kriging*“ →  $\boldsymbol{\mu}(\mathbf{s}) = m$   
 $m \dots$  wird dabei als bekannt vorausgesetzt. Somit wären auch die Werte von  $\boldsymbol{\varepsilon}(\mathbf{s})$  bekannt.
- „*Universal-Kriging*“ →  $\boldsymbol{\mu}(\mathbf{s}) = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 x^2 + \beta_4 y^2 + \beta_5 xy$   
 Hier kann  $\boldsymbol{\mu}(\mathbf{s})$  somit grundsätzlich einen beliebigen Trend aufweisen.  
 Daher ist auch die „*Universal-Kriging*“-Variante beim Vorliegen eines Trends der „*Ordinary-Kriging*“-Variante vorzuziehen!

Für die Wahl der jeweils anzuwendenden Variante ist es somit erforderlich zu bestimmen, ob ein räumlicher Trend vorhanden ist oder nicht. Falls kein Trend erkennbar ist wäre „*Ordinary-Kriging*“ das geeignete Interpolationsverfahren ansonsten sollte „*Universal-Kriging*“ verwendet werden.

Die allgemeine Berechnungsformel beim „Kriging“-Verfahren lautet wie folgt:

$$\hat{Z}(s_p) = \sum_{i=1}^n \lambda_i Z(s_i)$$

Quelle: EISANK & KOCH (2017b)

wobei:

- $i$  → die Werte von 1, 2, 3... bis n darstellen,
- $n$  → die Anzahl der Werte ist,
- $Z(s_i)$  → dem Messwert (Höhenwert) an der  $i$ -ten Position entspricht,
- $\lambda_i$  → die Gewichtung für den an der  $i$ -ten Position gemessenen Wert ist und
- $\hat{Z}(s_p)$  → dem vorhergesagten bzw. geschätzten Wert entspricht.

Die unbekannte Gewichtung  $\lambda_i$  hängt dabei von einem an die verwendeten Messpunkte angepassten Modell, der Entfernung zur vorhergesagten Position, und den räumlichen Beziehungen unter den um die vorhergesagte Position herum liegenden Messpunkten ab. Um mit der räumlichen Anordnung der Messpunkte und deren Gewichtungen arbeiten zu können müssen die statistischen Abhängigkeitswerte, also die räumliche Autokorrelation, quantifiziert werden.

Weiters gilt:

$$\left( Z(s_p) - \hat{Z}(s_p) \right)^2 \longrightarrow \min$$

Quelle: EISANK & KOCH (2017b)

wobei:

- $Z(s_p)$  → der tatsächliche Wert von P am Ort s ist und
- $\hat{Z}(s_p)$  → der geschätzte Wert von P am Ort s ist.

Um  $Z(s_p)$  zu ermitteln müssen daher die einzelnen  $\lambda_i$  – Werte mit Hilfe eines theoretischen Semivariogramms bestimmt werden. Hierfür stehen unterschiedliche Modelle zur Verfügung.

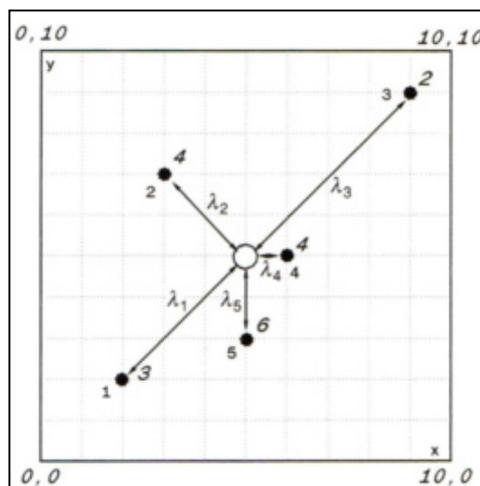


Abbildung 7: Beispiel für die Ermittlung von  $Z(s_p)$   
Quelle: BURROUGH & McDONNELL (1998)

Bei der „Ordinary Kriging“-Variante wird zwischen 5 Semivariogramm-Modellen unterschieden:

- „Spherical“

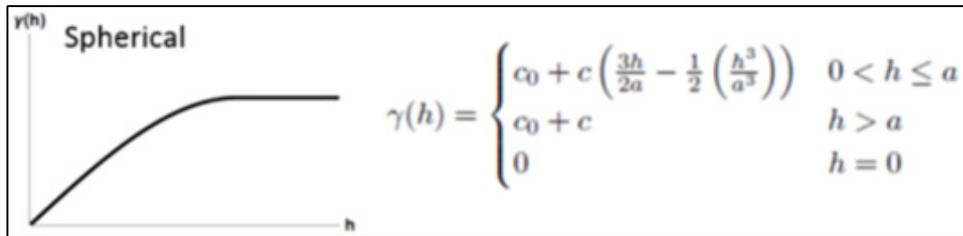


Abbildung 8: Semivariogramm-Modell – „Spherical“

Quelle: ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2016)

- „Circular“

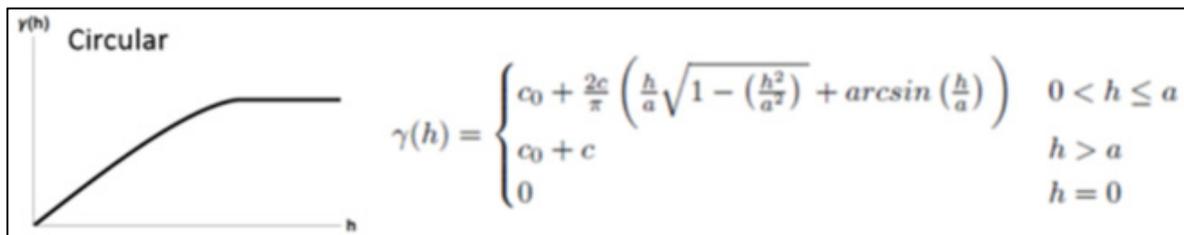


Abbildung 9: Semivariogramm-Modell – „Circular“

Quelle: ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2016)

- „Exponential“

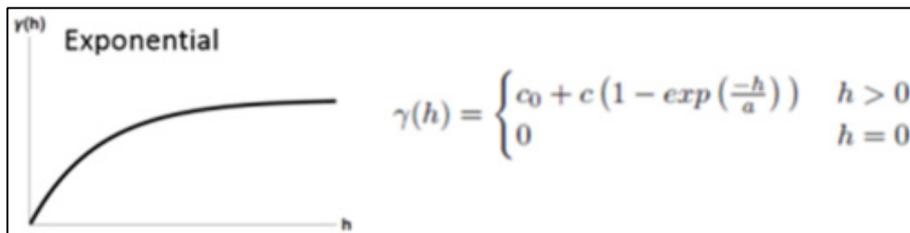


Abbildung 10: Semivariogramm-Modell – „Exponential“

Quelle: ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2016)

- „Gaussian“

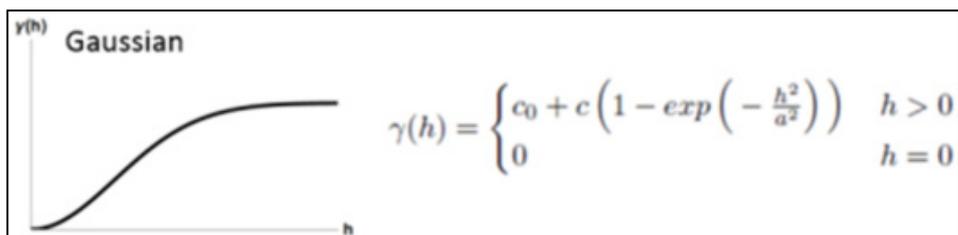


Abbildung 11: Semivariogramm-Modell – „Gaussian“

Quelle: ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2016)

▪ „Linear“

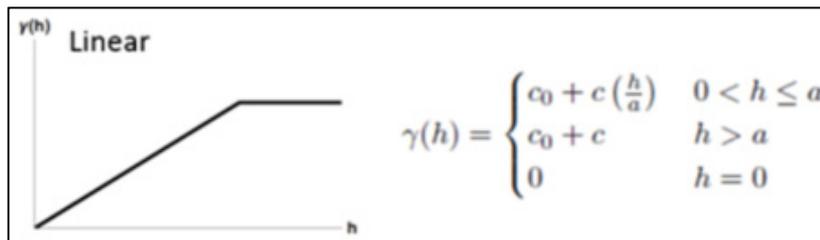


Abbildung 12: Semivariogramm-Modell – „Linear“

Quelle: ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2016)

Neben der Erstellung der eigentlichen Interpolationsoberfläche wird beim „Kriging“-Verfahren außerdem noch eine Kreuzvalidierung der Ergebnisse durchgeführt. Diese wird standardmäßig zur Beurteilung der Modellgüte von Interpolationsoberflächen herangezogen. Anhand einer Kreuzvalidierung (im englisch sprachlichen Raum als „Cross-Validation“ bekannt) können grundsätzlich unterschiedliche Interpolationsverfahren bzw. Methoden und deren Ergebnisse miteinander verglichen werden und somit das beste Resultat bzw. die beste Interpolationsvariante ermittelt werden. Hierfür werden u.a. unterschiedliche Fehlerwerte berechnet, welche sich Großteils auf das jeweilige Modell selbst bzw. auf beliebig viele Modelle beziehen, wenn mit den gleichen Daten gearbeitet wird. So gibt z.B. der „Average Standard Error“ den mittleren Fehler an den man macht wenn man unterschiedliche Modelle mit den jeweils verwendeten Daten berechnet (EISANK & KOCH 2017b).

Folgende Fehlerkennwerte werden bei der Kreuzvalidierung standardmäßig ermittelt:

- der „Mean Error“ („ME“),
- der „Root Mean Square Error“ („RMSE“),
- der „Mean Standardized Error“ („MSE“),
- der „Root Mean Square Standardized Error“ („RMSSE“) und
- der „Average Standard Error“ („ASE“).

Für die abschließende Beurteilung der Güte eines Interpolationsmodells wird dabei meist der „Mean Error“ und der „Root Mean Square Error“ herangezogen. Diese beiden Kennzahlen werden auch im Verlauf dieser Master Thesis als Teil der Kreuzvalidierung (siehe Kapitel 5.3 - Kreuzvalidierung) für alle Interpolationsoberflächen berechnet und anschließend graphisch und tabellarisch gegenübergestellt. Anhand dieser Fehlerkennwerte kann somit, neben einer visuellen und einer explorativen räumlichen Datenanalyse, eine fundierte Aussage hinsichtlich der Genauigkeit von unterschiedlichen Interpolationsoberflächen getroffen werden.

Vor der Durchführung einer Kreuzvalidierung erfolgt dabei meist eine visuelle Analyse der Interpolationsergebnisse. Bereits anhand dieser können, beim unmittelbaren Vergleich von unterschiedlichen Interpolationsverfahren bzw. Algorithmen, abweichende Muster festgestellt werden und damit für die anschließende Kreuzvalidierung markiert werden. Abbildung 13 zeigt zur Verdeutlichung die unterschiedlichen Interpolationsergebnisse (u.a. die „*Ordinary-Kriging*“-Variante mit den Semivariogramm-Modellen „*Exponential*“ bzw. „*Gaussian*“) hinsichtlich des von PAVLOVA (2017) durchgeführten Vergleichs von insgesamt 6 unterschiedlichen Interpolationsalgorithmen.

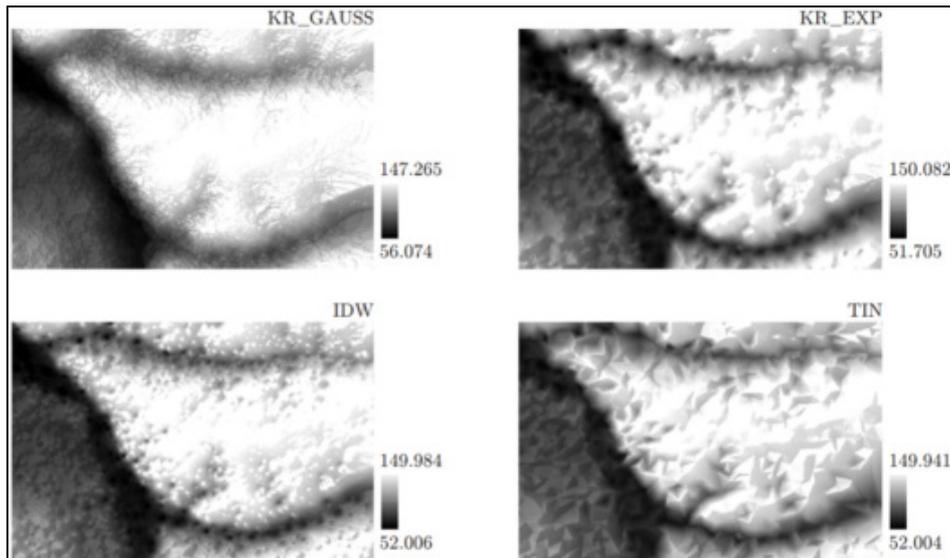


Abbildung 13: Visuelle Ergebnisse unterschiedlicher Interpolationsalgorithmen  
Quelle: PAVLOVA (2017)

Abbildung 14 und 15 zeigen ein Beispiel einer Kreuzvalidierung im „*Geostatistical Wizard*“ von ArcGIS Pro wobei hierbei eine Interpolationsfläche mittels „*Ordinary-Kriging*“ erstellt wurde.

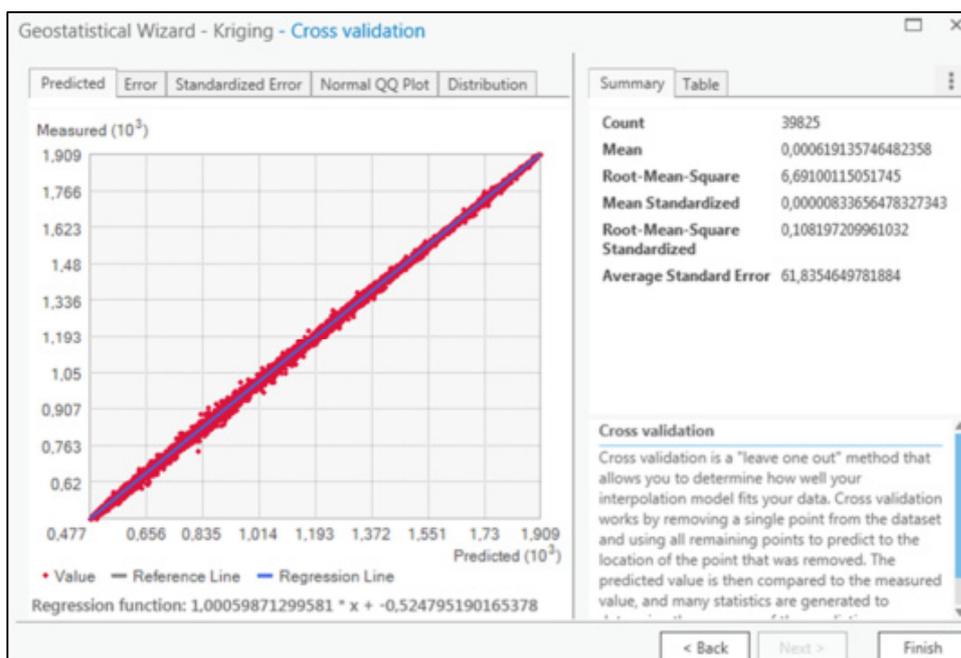


Abbildung 14: Beispiel einer Kreuzvalidierung - „*Ordinary-Kriging*“-Variante (Bild 1/2)

Neben der Berechnung der einzelnen Fehlerwerte (jeweils im Bild rechts unter „Summary“ dargestellt) werden beim „Kriging“-Interpolationsverfahren auch noch graphische Darstellungen (sogenannte „Scatterplots“) der, auf Basis der jeweils gewählten Interpolationsparameter, resultierenden Fehler (u.a. „Predicted“, „Error“, „Standardized-Error“) generiert. Das „Predicted“-Diagramm zeigt dabei das Verhältnis zwischen den gemessenen („Measured“) und den modellierten („Predicted“) Werten, wobei weiters auch die jeweilige Regressionsfunktion ermittelt und dargestellt wird.

Die nachfolgende Abbildung 15 stellt ergänzend dazu noch das allgemeine „Error“-Diagramm dar, also das Verhältnis zwischen den gemessenen („Measured“) Werten und dem jeweils auftretenden Fehler („Error“), wobei ebenfalls die daraus resultierende Regressionsfunktion berechnet wird.

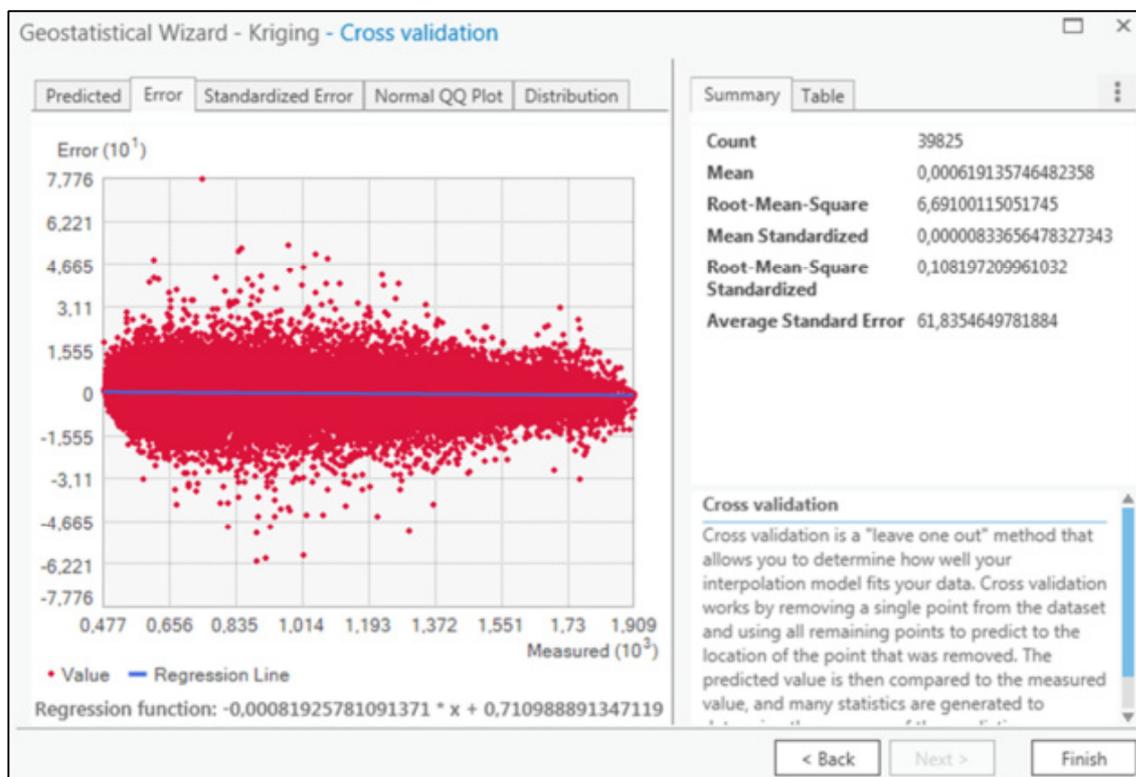


Abbildung 15: Beispiel einer Kreuzvalidierung - „Ordinary-Kriging“-Variante (Bild 2/2)

### 2.2.3 „SPLINE“

Beim „*Spline*“-Interpolationsverfahren handelt es sich wie beim „*IDW*“-Verfahren um eine deterministische Interpolationsmethode. Hierfür werden die Zellenwerte der zu erstellenden Interpolationsoberflächen basierend auf den umliegenden (Mess-)Punkten mit Hilfe einer mathematischen Funktion geschätzt, wobei die daraus resultierende Oberfläche jeweils exakt durch die verwendeten Eingabepunkte verläuft (BURROUGH & MCDONNELL 1998).

Es wird hierfür prinzipiell eine durch die (Mess-)Punkte verlaufende „Gummischeibe“ gebogen, wobei versucht wird die Gesamtkrümmung der berechneten Interpolationsoberfläche zu minimieren. Die mathematische Funktion wird dabei jeweils, entsprechend einer definierten Anzahl nächster Eingabepunkte, beim Verlauf durch die Referenzpunkte angepasst (ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE 2018e).

Bei diesem Interpolationsverfahren wird prinzipiell zwischen zwei Varianten unterschieden:

- **„Regularized-Spline“**

Dies ist die Standardvariante des „*Spline*“-Verfahrens und wird auch in der vorliegenden Master Thesis für den Vergleich der unterschiedlichen Interpolationsverfahren verwendet. Mit Hilfe dieser Variante wird grundsätzlich eine glatte Ergebnisoberfläche erzeugt, wobei die berechneten Zellenwerte auch außerhalb des Referenzdatenbereichs liegen können. Der Gewichtungparameter  $T$  („*Tau*“ aus dem griechischen Alphabet) gibt dabei die Gewichtung der Werte der dritten Ableitung während der Minimierung an. Je höher dieser Hauptingabeparameter gewählt wird desto „glattere“ Interpolationsoberflächen entstehen (ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE 2018e).

- **„Tension-Spline“**

Hier wird die Steifigkeit der zu erstellenden Interpolationsoberfläche gemäß dem Charakter des modellierten Phänomens gesteuert, wobei als Resultat eine weniger glatte Oberfläche erzeugt wird. Die berechneten Zellenwerte sind grundsätzlich durch den Referenzdatenbereich stärker eingeschränkt. Der Gewichtungparameter  $\Phi$  („*Phi*“ aus dem griechischen Alphabet) gibt dabei die Gewichtung der Werte der ersten Ableitung während der Minimierung an. Je höher dieser Parameter gewählt wird desto „gröbere“ Interpolationsoberflächen entstehen (ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE 2018e). Diese Variante eignet sich des Weiteren gut zur Verarbeitung von großen Datensätzen, wobei die Genauigkeit hinsichtlich der Interpolationsergebnisse nahe bei dem mancher „*Kriging*“-Methoden liegt (XIAO et al. 2016).

Beim „Spline“-Interpolationsalgorithmus erfolgt die Oberflächeninterpolation grundsätzlich anhand folgender Berechnungsformel:

$$S(x,y) = T(x,y) + \sum_{j=1}^N \lambda_j R(r_j)$$

Quelle: ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2018e)

wobei:

- $j$  → die Werte von 1, 2, 3... bis N darstellen,
- $N$  → die Anzahl der Punkte ist,
- $\lambda_j$  → sind Koeffizienten (durch Lösung eines Systems linearer Gleichungen),
- $r_j$  → ist die jeweilige Entfernung vom Punkt (X,Y) zum j -ten Punkt,
- $T(x,y)$  → abhängig von der jeweiligen „Spline“-Variante ist,
- $R(r)$  → ist ebenfalls abhängig von der „Spline“-Variante.

Für die erste Variante „Regularized-Spline“ gilt dabei:

$$T(x,y) = a_1 + a_2x + a_3y$$

Quelle: ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2018e)

und

$$R(r) = \frac{1}{2\pi} \left\{ \frac{r^2}{4} \left[ \ln \left( \frac{r}{2\tau} \right) + c - 1 \right] + \tau^2 \left[ K_0 \left( \frac{r}{\tau} \right) + c + \ln \left( \frac{r}{2\pi} \right) \right] \right\}$$

Quelle: ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2018e)

wobei:

- $a_i$  → sind Koeffizienten (durch Lösung eines Systems linearer Gleichungen),
- $r$  → die Entfernung zwischen dem Punkt und der Stichprobe ist,
- $\tau$  → den „Weight“-Parameter beim „Regularized-Spline“-Typ darstellt,
- $K_0$  → die modifizierte Bessel-Funktion ist,
- $c$  → als Konstante mit einem Wert von 0,577215 definiert wird.

Die nachfolgende Abbildung 16 zeigt als Beispiel eine Interpolationsoberfläche welche mit der Variante „Regularized-Spline“ und einem „Weight“-Parameter von 0,5 erstellt wurde:

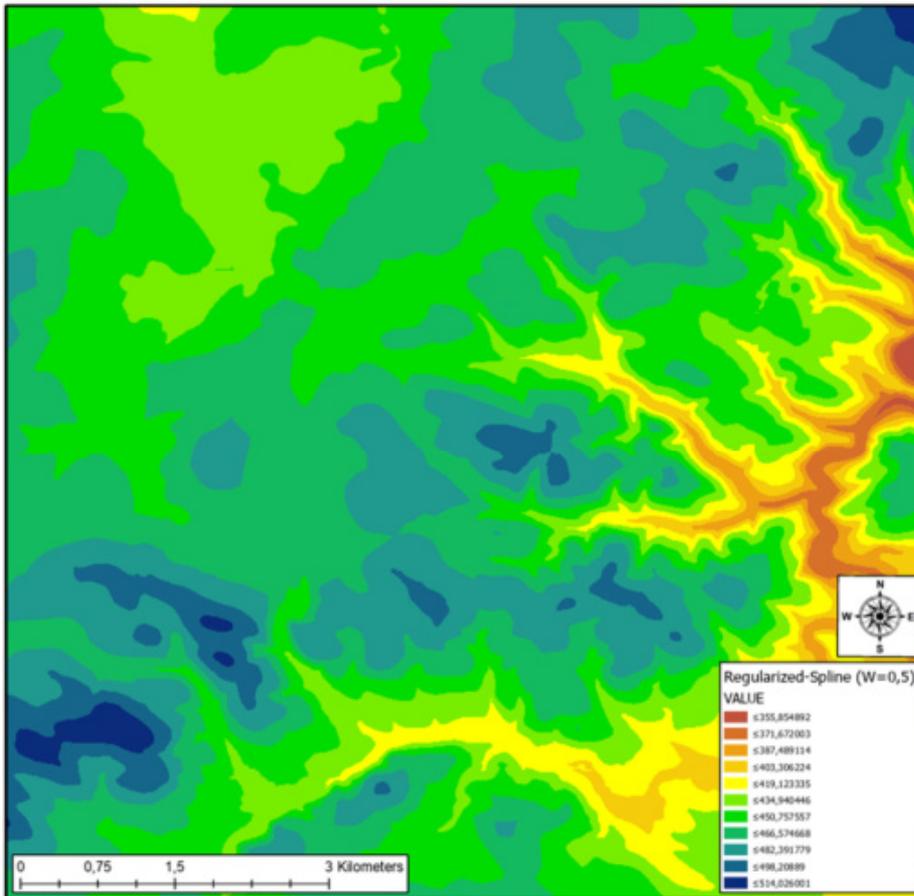


Abbildung 16: „Regularized-Spline“-Interpolationsoberfläche mit „Weight“ = 0,5

Für die zweite Variante „Tension-Spline“ gilt dabei:

$$T(x,y) = a_1$$

Quelle: ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2018e)

und

$$R(r) = - \frac{1}{2\pi\phi^2} \left[ \ln\left(\frac{r\phi}{2}\right) + c + K_0(r\phi) \right]$$

Quelle: ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2018e)

wobei:

- $a_1$  → ist ein Koeffizient (durch Lösung eines Systems linearer Gleichungen),
- $r$  → die Entfernung zwischen dem Punkt und der Stichprobe ist,
- $\varphi^2$  → den „*Weight*“-Parameter beim „*Tension-Spline*“-Typ darstellt,
- $K_0$  → die modifizierte Bessel-Funktion ist,
- $c$  → wiederum als Konstante mit einem Wert von 0,577215 definiert wird.

Die nachfolgende Abbildung 17 zeigt als Beispiel eine Interpolationsoberfläche welche mit der Variante „*Tension-Spline*“ und einem „*Weight*“-Parameter von 5,0 erstellt wurde:

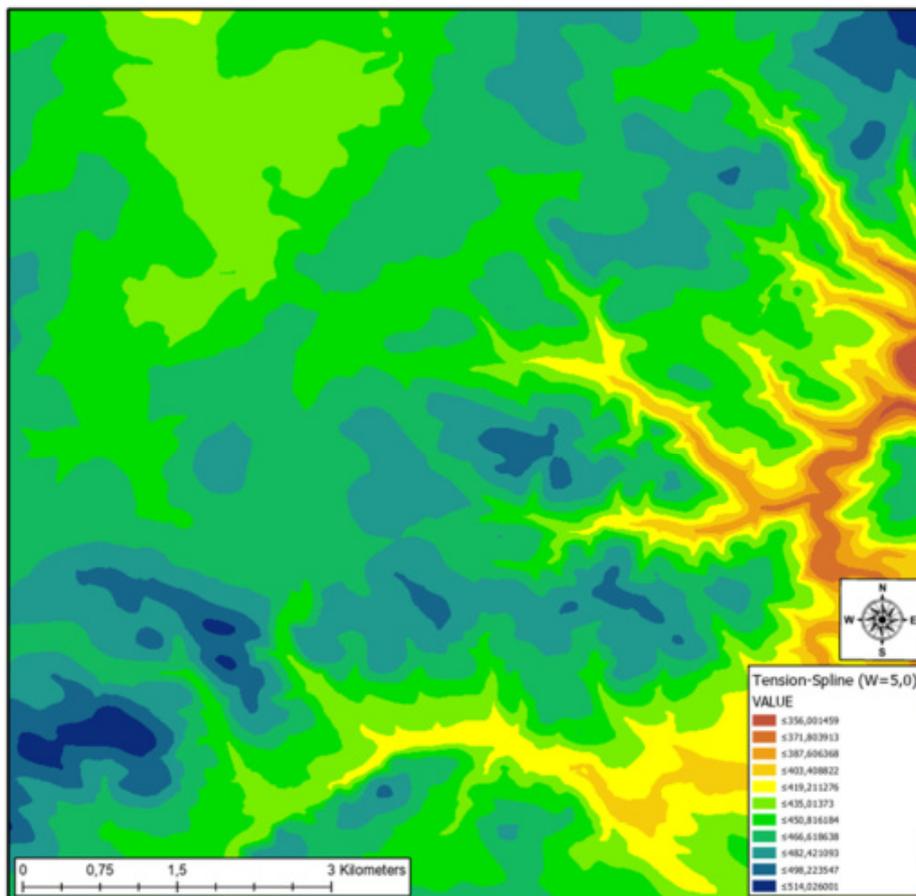


Abbildung 17: „*Tension-Spline*“-Interpolationsoberfläche mit „*Weight*“ = 5,0

Beide „*Spline*“-Varianten stehen dabei grundsätzlich in allen gängigen GIS-Softwareprodukten für die Berechnung von Interpolationsoberflächen zur Verfügung. Somit ist dieses Interpolationsverfahren und deren zugrundeliegender (Geo-)Algorithmus unabhängig der verwendeten GIS-Applikation für eine Mehrheit potentieller Anwender interessant, wobei die Wahl der für den jeweiligen Anwendungsfall geeigneten „*Spline*“-Variante beachtet und eine (Kreuz-)Validierung der daraus resultierenden Ergebnisse durchgeführt werden sollte.

## 2.3 Sonstige für den Geoprozess verwendete (Geo-)Algorithmen

Der zum Erstellen der Interpolationsoberflächen und zum Validieren der Ergebnisse verwendete Geoprozess beinhaltet noch weitere (Geo-)Algorithmen, wobei auf diese, bis auf einer kurzen nachfolgenden theoretischen Beschreibung, in der vorliegenden Arbeit nicht näher eingegangen wird:

### „**PROJECT RASTER**“

Dieser Interpolationsalgorithmus unterstützt Koordinatentransformationen zwischen zwei beliebigen Projektionen. Hierfür wird ein Quell-Raster („*In-Raster*“) samt Projektion benötigt welches, nach der Definition aller notwendigen Parameter (des Eingabe- bzw. Ausgabe-Koordinatensystems, der geographischen Transformationsmethode, der „*Resampling*“-Technik, der Ausgabe-Zellgröße und ggf. optionalen Parametern) als neues Ziel-Raster („*Out-Raster*“) erstellt und anschließend gespeichert wird. Somit entsteht ein eigenes Raster-Dataset mit dem gewünschten Raumbezug, wobei hierfür meist eine auf bilineare Interpolation beruhende Näherungsmethode verwendet wird (ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE 2018d).

### „**CONTOUR**“

Der hierfür verwendete Algorithmus wird eingesetzt um ein „*Polyline-Feature-Dataset*“ aus einem (Quell-)Raster zu erstellen. Dabei werden (Poly-)Linien erzeugt, die Positionen desselben Wertes verbinden. Angewendet wird dieser (Geo-)Algorithmus grundsätzlich für kontinuierliche Phänomene wie Höhe, Temperatur, Niederschlag, Luftdruck etc. Die Linien-Features verbinden dabei alle Zellen eines konstanten Werts in der Eingabe. Die dadurch resultierende Verteilung der Konturlinien zeigt, wie sich Werte über eine Oberfläche hinweg verändern. Ist z.B. der Abstand zwischen den einzelnen Linien groß deutet dies auf eine geringe Veränderung der Werte hin. Bei stark variierenden Werten ist hingegen der Abstand zwischen den Linien geringer (ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE 2018a). Mit Hilfe dieses Algorithmus wird somit anhand eines (Quell-)Rasters als Eingangsdatenquelle als Resultat ein „*Vector-Dataset*“ mit Konturlinien erstellt.

### „**INTERSECT**“

Der hierfür verwendete Algorithmus berechnet den geometrischen Schnittpunkt von „*Feature-Classes*“ bzw. „*Feature-Layern*“, wobei die diesbezügliche Anzahl beliebig groß sein kann. In die Ergebnis „*Feature-Class*“ werden lediglich jene „*Features*“ aufgenommen, welche in allen verwendeten Eingaben vorkommen. Grundsätzlich werden dabei zuerst die geometrischen Beziehungen zwischen den einzelnen Eingabe „*Feature-Layern*“ analysiert, und dann die dadurch ermittelten Überschneidungen anschließend wahlweise als Punkte, Linien oder Polygone in die Ausgabe „*Feature-Class*“ geschrieben. Mit Hilfe des „Ausgabetypp“-Parameters wird definiert, welcher „*Feature*“-Typ als Ergebnis der Überschneidungen gespeichert wird (ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE 2018c).

### 3. Basisdatensätze

#### 3.1 Datengrundlage

Für den Vergleich der drei gängigsten Interpolationsalgorithmen und der Analyse der damit zu erstellenden Interpolationsoberflächen, werden insgesamt drei verschiedene Basisdatensätze als Eingangsdaten(quellen) verwendet. Die ersten beiden Basisdatensätze stammen dabei vom Bundesamt für Eich- und Vermessungswesen (BEV). Geographisch gesehen handelt sich bei diesen räumlichen Daten um ein Teilgebiet von Villach im Bundesland Kärnten in Österreich. Beide Datensätze können unentgeltlich als sogenannte DGM-Höhenraster Produktbeispiele von der BEV-Homepage ([http://www.bev.gv.at/portal/page?\\_pageid=713,1573009&\\_dad=portal&\\_schema=PORTAL](http://www.bev.gv.at/portal/page?_pageid=713,1573009&_dad=portal&_schema=PORTAL)) bezogen werden.

Die folgenden Abbildungen 18 und 19 in Form einer ÖK50-Übersichtskarte und eines Orthofotos geben einen groben Überblick bezüglich der räumlichen Ausprägung der ersten beiden Basisdatensätze. Das Orthofoto bildet dabei das exakte Teilgebiet der verwendeten BEV-Basisdatensätze ab.



Abbildung 18: ÖK50 mit Detail-Orthofoto – Grenzen der Basisdaten (grüne umrandet)



Abbildung 19: Detail-Orthofoto – Räumliche Ausdehnung / Grenzen der BEV-Basisdaten

Die in Abbildung 18 verwendete staatliche österreichische Karte (BEV-Bezeichnung: „0002311146\_300\_KM50-R.jpg“) wird aus dem kartographischen Raster-Modell 1:50.000 abgeleitet und steht ebenso wie das in Abbildung 19 verwendete Orthofoto (BEV-Bezeichnung: „0000475208\_100\_OF\_4917-48.tif“) als unentgeltliches Produktbeispiel auf der BEV-Homepage frei zum Download zur Verfügung.

Bei den ersten beiden Basisdatensätzen handelt es sich jeweils um DGM-Höhenraster in Form von „DXF-Gridpoints“ mit einer Rasterweite von 5x5 Meter (Basisdatensatz 1) bzw. 25x25 Meter (Basisdatensatz 2). Das räumliche Bezugs- und Koordinatensystem, der EPSG-Code sowie das Höhenbezugssystem ist dabei bei beiden Datensätzen ident. Auch die Position und räumliche Ausdehnung von 1250 Meter in horizontaler Ebene und 1000 Meter in vertikaler Ebene sind konstant. Somit unterscheiden sich die in weiterer Folge verwendeten beiden BEV-Basisdatensätze hauptsächlich durch deren unterschiedliche Rasterweite von 5x5 bzw. 25x25 Meter. Dadurch variiert bei diesen beiden Datensätzen die Datendichte stark. Beim 5x5 Meter Raster beträgt die Summe der im weiteren Verlauf verwendeten Rasterpunkte bei ca. 50.000 Punkten. Beim 25x25 Meter Raster liegt die Summe lediglich bei ca. 2.100 Punkten und weist somit eine deutlich niedrigere Datendichte auf.

### 3.2 Basisdatensatz 1

- **Allgemeine Informationen bezüglich des Datensatzes:**

Quelle:	Bundesamt für Eich- und Vermessungswesen (BEV)
Dateiname:	4917-48_RG5.dxf
Datentyp:	DGM-Höhenraster
Datenformat:	<i>DXF-Gridpoints</i>
Rasterweite:	5 x 5 Meter (als gleichmäßige Punkteverteilung)
Räumliches Bezugssystem:	MGI - GK M31 ( <i>Austria GK Central</i> )
EPSG-Code:	31255
Kartenprojektion:	Gauß-Krüger
Geodätisches Datum:	Militärgeographisches Institut (MGI)
Referenzellipsoid:	Bessel 1841
Höhenbezugssystem:	Adria Triest
Räumliche Ausdehnung:	Rechtswert min.: 38.750 m Rechtswert max.: 40.000 m Hochwert min.: 164.000 m Hochwert max.: 165.000 m
Datenreferenz / Downloadlink:	<a href="http://www.bev.gv.at/portal/page?_pageid=713,2155737&amp;_dad=portal&amp;_schema=PORTAL">http://www.bev.gv.at/portal/page?_pageid=713,2155737&amp;_dad=portal&amp;_schema=PORTAL</a>

Tabelle 2: Grundlegende Informationen – Basisdatensatz 1

- **Trendanalyse / Trendoberfläche:**

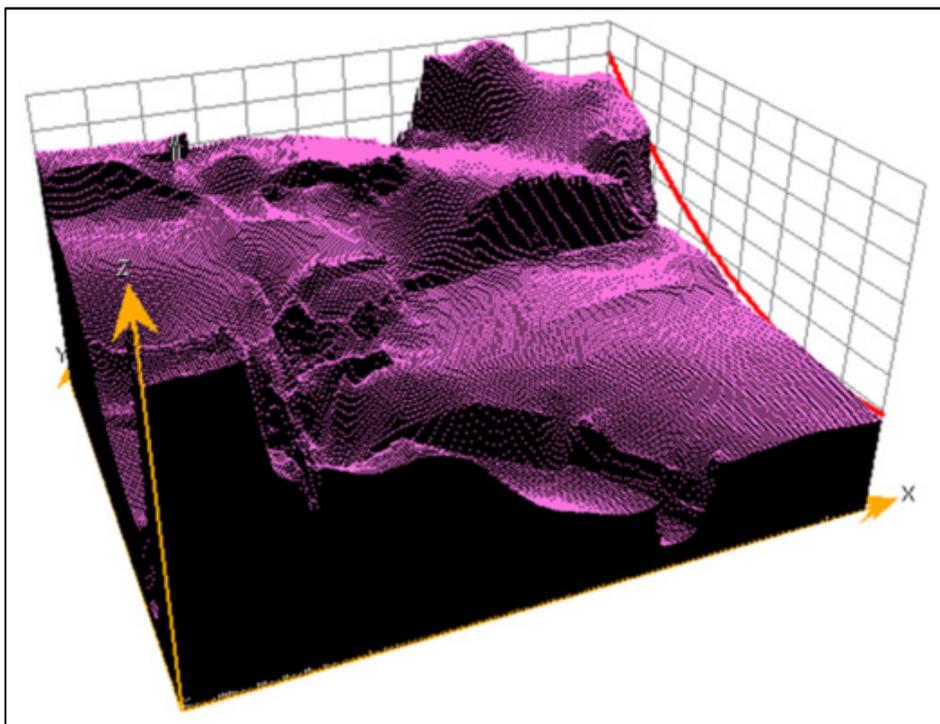


Abbildung 20: Trendoberfläche – Basisdatensatz 1

Abbildung 20 zeigt die Trendoberfläche für den Basisdatensatz 1. Dabei werden die gegebenen Z-Werte (Attributname: „Elevation“) in Form einer dreidimensionalen Darstellung über den X- und Y-Koordinaten dargestellt. Damit können mögliche Trends ermittelt werden, wobei die rote Kurve (in der rechten äußeren Gitterfläche dargestellt) einen relativ linearen Trend zunehmender Höhe von Süd in Richtung Nord andeutet.

• **Deskriptive stat. Maße und Werteverteilung für das Höhenattribut „Elevation“:**

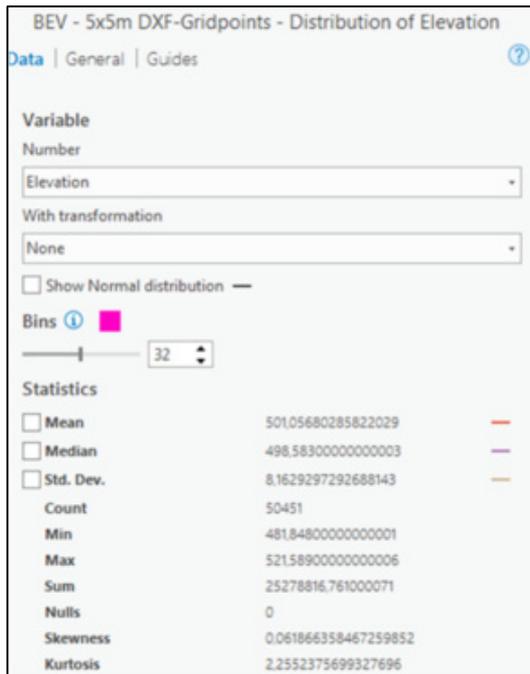


Abbildung 21: Deskriptive stat. Maße – Basisdatensatz 1

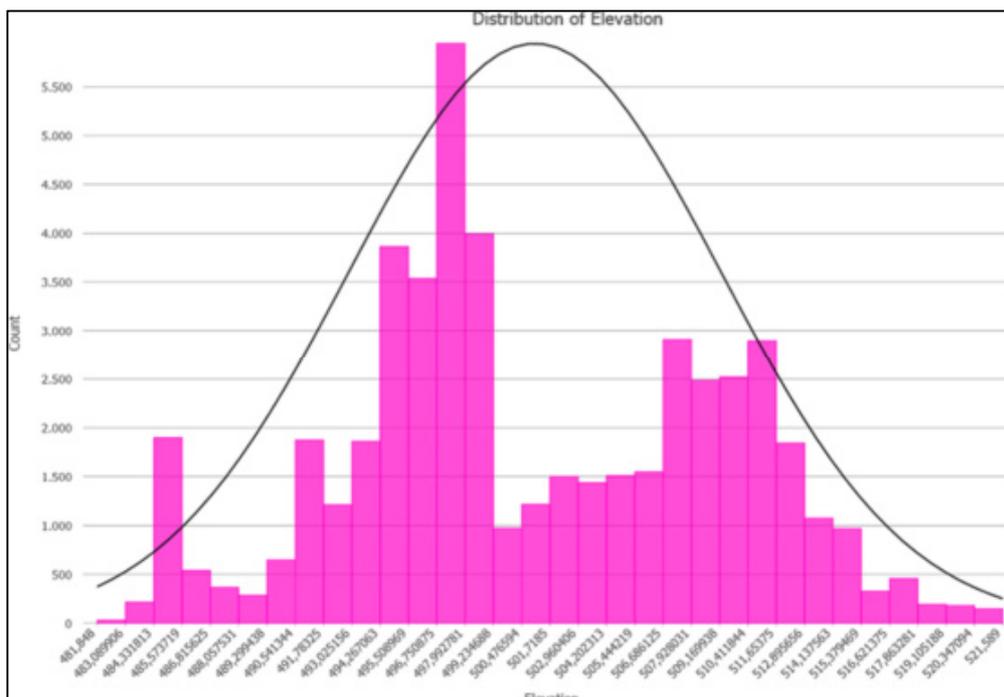


Abbildung 22: „Elevation“-Höhenattribut Histogramm – Basisdatensatz 1

### 3.3 Basisdatensatz 2

- **Allgemeine Informationen bezüglich des Datensatzes:**

Quelle:	Bundesamt für Eich- und Vermessungswesen (BEV)
Dateiname:	4917-48_RG25.dxf
Datentyp:	DGM-Höhenraster
Datenformat:	<i>DXF-Gridpoints</i>
Rasterweite:	25 x 25 Meter (als gleichmäßige Punkteverteilung)
Räumliches Bezugssystem:	MGI - GK M31 ( <i>Austria GK Central</i> )
EPSG-Code:	31255
Kartenprojektion:	Gauß-Krüger
Geodätisches Datum:	Militärgeographisches Institut (MGI)
Referenzellipsoid:	Bessel 1841
Höhenbezugssystem:	Adria Triest
Räumliche Ausdehnung:	Rechtswert min.: 38.750 m Rechtswert max.: 40.000 m Hochwert min.: 164.000 m Hochwert max.: 165.000 m
Datenreferenz / Downloadlink:	<a href="http://www.bev.gv.at/portal/page?_pageid=713,2155737&amp;_dad=portal&amp;_schema=PORTAL">http://www.bev.gv.at/portal/page?_pageid=713,2155737&amp;_dad=portal&amp;_schema=PORTAL</a>

Tabelle 3: Grundlegende Informationen – Basisdatensatz 2

- **Trendanalyse / Trendoberfläche:**

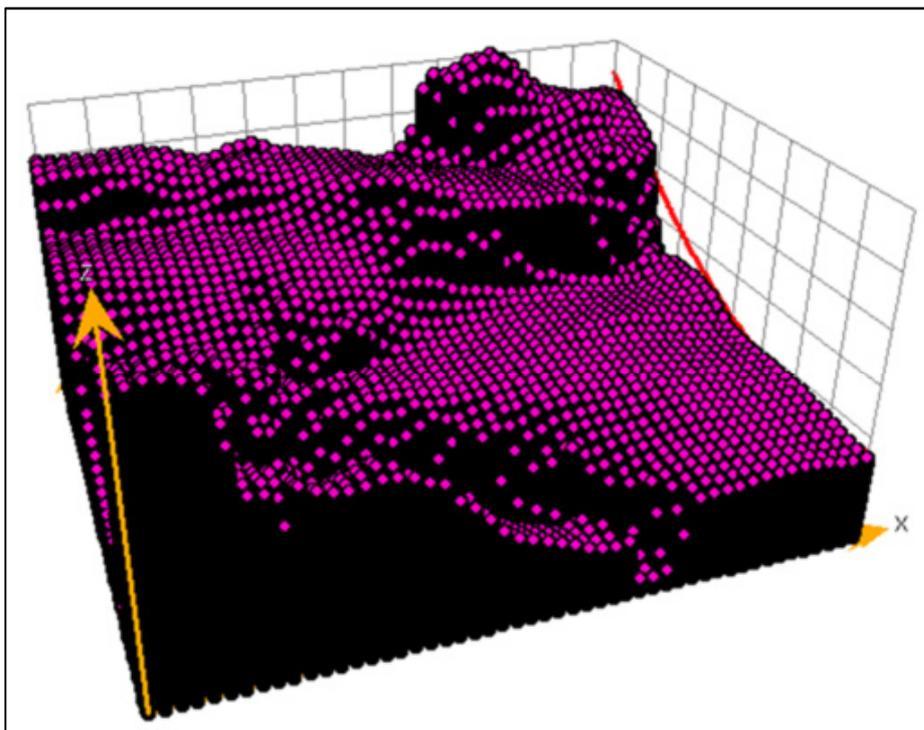


Abbildung 23: Trendoberfläche – Basisdatensatz 2

Abbildung 23 zeigt die Trendoberfläche für den Basisdatensatz 2. Dabei werden wiederum die gegebenen Z-Werte (Attributname: „Elevation“) in Form einer dreidimensionalen Darstellung über den X- und Y-Koordinaten dargestellt. Damit können mögliche Trends ermittelt werden, wobei die rote Kurve (in der rechten äußeren Gitterfläche dargestellt) wiederum einen relativ linearen Trend zunehmender Höhe von Süd in Richtung Nord andeutet.

• **Deskriptive stat. Maße und Werteverteilung für das Höhenattribut „Elevation“:**

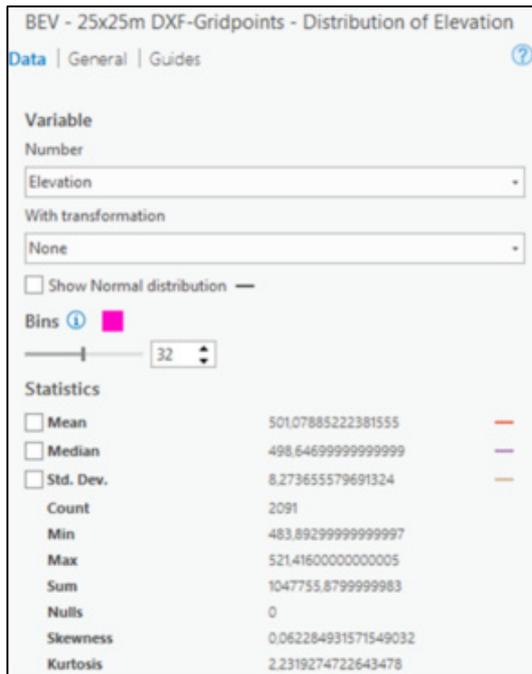


Abbildung 24: Deskriptive stat. Maße – Basisdatensatz 2

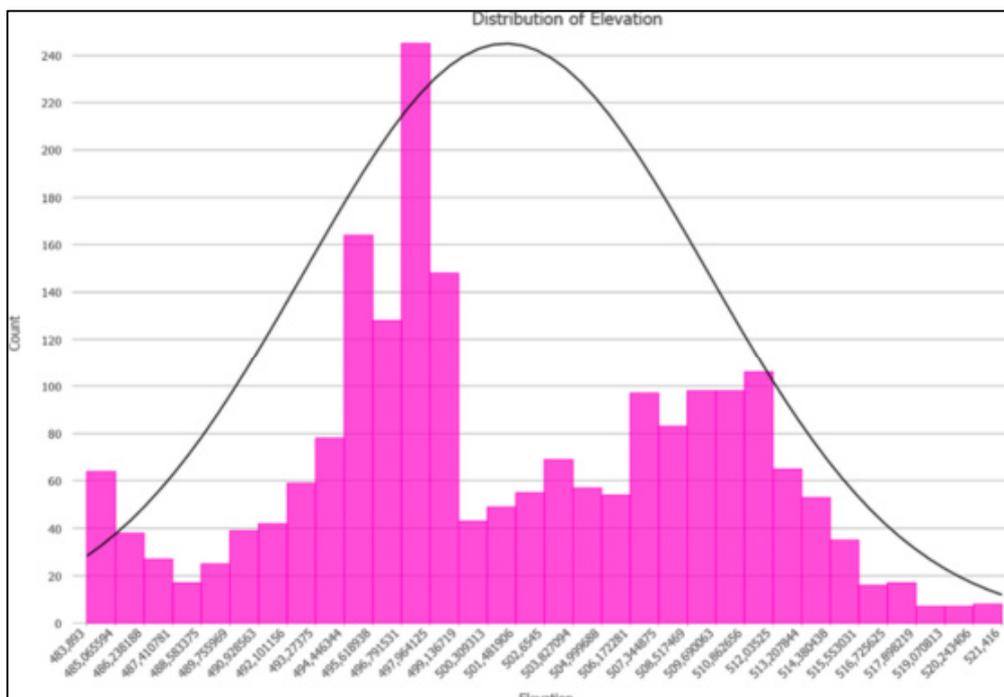


Abbildung 25: „Elevation“-Höhenattribut Histogramm – Basisdatensatz 2



- **Allgemeine Informationen bezüglich des Datensatzes:**

Quelle:	CGIAR Consortium for Spatial Information (CGIAR-CSI)
Dateiname:	SRTM_39_03.tif
Datentyp:	SRTM 90m DEM Version 4
Datenformat:	<i>SHP-Gridpoints</i> (Punkte aus Geo-TIFF Datei extrahiert)
Rasterweite:	78 x 78 Meter (als gleichmäßige Punkteverteilung)
Räumliches Bezugssystem:	MGI - GK M31 ( <i>Austria GK Central</i> )
EPSG-Code:	31255
Kartenprojektion:	Gauß-Krüger
Geodätisches Datum:	Militärgeographisches Institut (MGI)
Referenzellipsoid:	Bessel 1841
Höhenbezugssystem:	Adria Triest
Räumliche Ausdehnung:	Rechtswert min.: ca. 30.495 m Rechtswert max.: ca. 48.054 m Hochwert min.: ca. 162.802 m Hochwert max.: ca. 176.598 m
Datenreferenz / Downloadlink:	<a href="https://cgiaresci.community/data/srtm-90m-digital-elevation-database-v4-1/">https://cgiaresci.community/data/srtm-90m-digital-elevation-database-v4-1/</a>

Tabelle 4: Grundlegende Informationen – Basisdatensatz 3

- **Deskriptive stat. Maße und Werteverteilung für das Höhenattribut „Elevation“:**

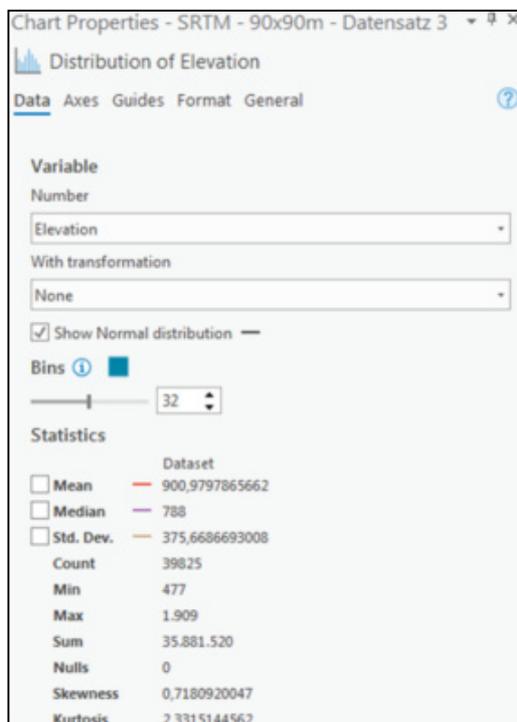


Abbildung 27: Deskriptive stat. Maße – Basisdatensatz 3

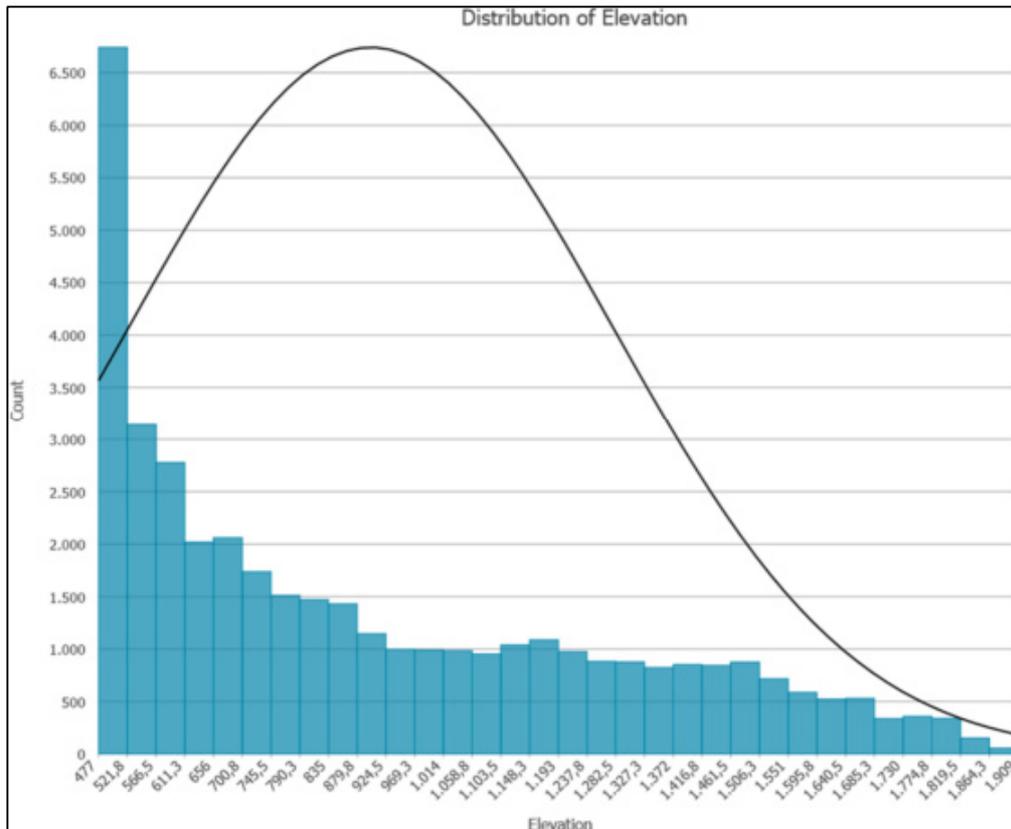


Abbildung 28: „Elevation“-Höhenattribut Histogramm – Basisdatensatz 3

- **Trendanalyse / Trendoberfläche:**

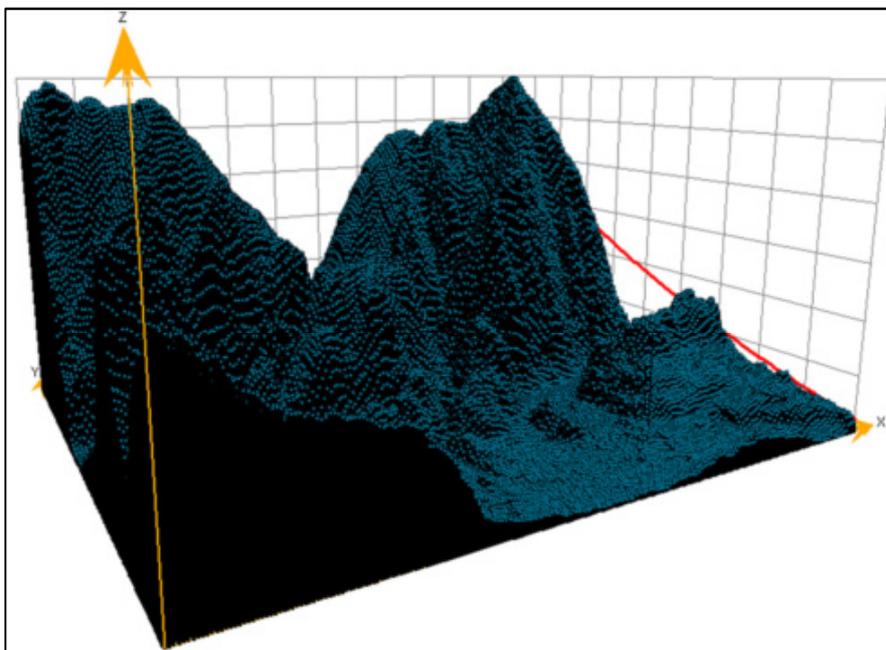


Abbildung 29: Trendoberfläche – Basisdatensatz 3

Die Trendoberfläche für diesen dritten Basisdatensatz deutet anhand der roten Kurve wiederum einen relativ linearen Trend zunehmender Höhe von Süd in Richtung Nord an. Des Weiteren unterscheidet sich die Struktur der Oberfläche deutlich von der Struktur der beiden BEV-Datensätze.

## 3.5 Zusammengefasste Charakteristika der drei Basisdatensätze

Charakteristika	Basisdatensatz 1	Basisdatensatz 2	Basisdatensatz 3
Untersuchungsgebiet:	Villach (Teilgebiet)	Villach (Teilgebiet)	Kärnten (Teilgebiet)
Datenart (real/experimentell):	reale Daten	reale Daten	reale Daten
Datenursprung:	BEV *	BEV *	SRTM **
Datentyp:	Höhenrasterpunkte	Höhenrasterpunkte	Höhenrasterpunkte
Datenformat:	<i>DXF-Gridpoints</i>	<i>DXF-Gridpoints</i>	<i>SHP-Gridpoints</i>
Punkteabstand:	5 x 5 Meter	25 x 25 Meter	78 x 78 Meter
Datendichte:	sehr hoch	mittel	gering
Räumliches Bezugssystem:	<i>Austria GK Central</i>	<i>Austria GK Central</i>	<i>Austria GK Central</i>
Kartenprojektion:	Gauß-Krüger	Gauß-Krüger	Gauß-Krüger
Geodätisches Datum:	MGI	MGI	MGI
EPSG-Code:	31255	31255	31255
Referenzellipsoid:	Bessel 1841	Bessel 1841	Bessel 1841
Höhenbezugssystem:	Adria Triest	Adria Triest	Adria Triest
Räumliche Ausdehnung X:	1.250 Meter	1.250 Meter	ca. 17.500 Meter
Räumliche Ausdehnung Y:	1.000 Meter	1.000 Meter	ca. 13.800 Meter
Gesamtfläche-Datensatz:	1,25 km <sup>2</sup>	1,25 km <sup>2</sup>	ca. 241,5 km <sup>2</sup>
Anzahl der Datenpunkte:	50.451	2.091	39.825
Topographie:	flaches Gelände mit geringen Höhenunterschieden (ca. 481m bis 521m)	flaches Gelände mit geringen Höhenunterschieden (ca. 483m bis 521m)	unregelmäßiges Gelände mit teils großen Höhenunterschieden (477m bis 1.909m)
Minimum aller Punktwerte:	481,85 Meter	483,89 Meter	477,00 Meter
Maximum aller Punktwerte:	521,59 Meter	521,41 Meter	1.909,00 Meter
Mittelwert:	501,06 Meter	501,08 Meter	900,98 Meter
Median bzw. Zentralwert:	498,58 Meter	498,65 Meter	788,00 Meter
Standardabweichung:	8,16	8,27	375,67
Symmetrie / Asymmetrie:	0,06	0,06	0,72
Wölbung / Kompaktheit:	2,26	2,23	2,33
Nearest Neighbor Ratio:	ca. 2,0 ***	ca. 2,0 ***	ca. 2,0 ***
z-Score:	ca. 433,27 ***	ca. 89,93 ***	ca. 385,32 ***
p-Value:	0 ***	0 ***	0 ***
Punkteverteilung:	<i>Dispersed</i> ***	<i>Dispersed</i> ***	<i>Dispersed</i> ***
* unentgeltlicher Datensatz vom Bundesamt für Eich- und Vermessungswesen			
** "Shuttle Radar Topography Mission"-Datensatz (SRTM 90m DEM Version 4)			
*** lt. durchgeführter "Nearest Neighbor"-Analysen → im Anhang D enthalten			

Tabelle 5: Zusammengefasste Charakteristika – Basisdatensatz 1 bis 3

## 4. Geoprozess

### 4.1 Spezifikation und Anforderungen

Wie bereits in Kapitel 1.3 beschrieben wurde, wird ein automatisierter Geoprozess entworfen und programmiert mit welchem die drei Interpolationsalgorithmen getestet, alle entsprechenden Interpolationsoberflächen berechnet, und diese anschließend gegenübergestellt und analysiert werden.

Zur Planung von Windparks in Österreich sowie im gesamten europäischen Raum wird von den zuständigen Planungs- und Ingenieurbüros häufig eine Software namens WindPRO eingesetzt. Hersteller und Entwickler dieser Applikation ist das Unternehmen EMD International A/S mit Stammsitz in Dänemark. Es handelt sich dabei um ein modular aufgebautes Programmsystem zur Projektierung von einzelnen Windenergieanlagen (WEAs) bis hin zu gesamten Windparks. Mit Hilfe von WindPRO können u.a. die zu erwartende Energieproduktion, die Lärmbelastung für die unmittelbaren Wohnnachbarschaftsgebiete, die Wirtschaftlichkeit sowie die visuellen Auswirkungen in Form von Sichtbarkeitsanalysen berechnet und visualisiert werden. Dabei arbeitet das Programm nach dem Prinzip der objektorientierten Projektierung. Jedes Windenergieprojekt besteht aus einer Vielzahl von verschiedenen Objekten wie z.B. WEAs, Schall-Immissionsorten, Schattenrezeptoren, Hindernissen sowie Windstatistiken bzw. Prognosen in Form von „METEO“ bzw. Terraindatenobjekten (NIELSEN 2010). Ähnlich wie bei den ESRI ArcGIS-Produkten, wo die Funktionalität mit Hilfe von „*Extensions*“ sukzessive erweitert werden kann, besteht WindPRO ebenfalls aus einer Basisapplikation und einzelnen Berechnungsmodulen. Diese unterscheiden sich je nach Art der durchzuführenden Berechnung und können einzeln erworben und lizenziert werden. Wird nun in WindPRO ein neues Projekt gestartet, werden neben den Geländerauigkeiten, georeferenzierten Hintergrundkarten (topographische Karten, Orthofotos etc.) auch exakte Höheninformationen für das spezifische Projektgebiet benötigt. Entweder können hier direkt Onlinedaten („*SRTM*“-Daten mit einer niedrigen Auflösung von 90 x 90 Meter) geladen werden, oder es werden Höhenlinien im DXF- bzw. SHP-Format importiert. Diese werden anschließend zur Erstellung des Höhenprofils für das Projektgebiet verwendet. Für Windenergieprojekte in Österreich werden in diesem Zusammenhang grundsätzlich BEV-Daten verwendet, und aus Kostengründen als DGM-Höhenraster in Form von „*DXF-Gridpoints*“ erworben. Dies entspricht exakt den Charakteristika der drei Basisdatensätze, welche in Kapitel 3 dieser Arbeit vorgestellt wurden.

Um diese Daten in WindPRO verwenden zu können sind jedoch noch mehrere Arbeitsschritte notwendig. Hierfür wird ein geographisches Informationssystem benötigt, wobei sich die Automatisierung der notwendigen Schritte mit Hilfe der Geo-Applikationsentwicklung nahezu aufdrängt.

Anhand einer benutzerspezifischen Erweiterung in Form eines Python Add-Ins können die hierfür notwendigen, und in allen gängigen GIS-Applikationen implementierten (Geo-)Algorithmen sowie deren notwendigen (Eingabe-)Parameter automatisiert verwendet werden.

Der strukturelle Ablauf des Geoprozesses definiert sich dabei wie folgt:

1. Berechnung der Interpolationsoberflächen aus den gegebenen Punkt(mess)daten (Basisdatensatz 1 bis 3) mit Hilfe der Interpolationsmethoden „IDW“, „Kriging“ und „Spline“.
2. Je berechneter Interpolationsoberfläche einer Änderung der räumlichen Projektion von MGI – GK M31 („Austria GK Central“) zu ETRS1989 – UTM Zone 33N.
3. Erstellung von Kontur- bzw. Höhenlinien aus den zuvor umprojizierten Interpolationsoberflächen für die gesamte räumliche Ausdehnung der verwendeten Basisdatensätze 1 bis 3.
4. Das Erzeugen von exakt definierten räumlichen Ausschnitten aller Höhenlinienmodelle mit Hilfe eines begrenzenden Polygons („Limitation-Area.shp“), wobei die daraus resultierenden Ausschnitte als Ergebnis exakt dem jeweiligen Windpark-Projektgebiet entsprechen.

Das Resultat sind somit einerseits die für den Vergleich der drei Interpolationsverfahren benötigten Berechnungsergebnisse in Form von multiplen Interpolationsoberflächen und andererseits die erzeugten Endergebnisse, welche bei Bedarf für die Erstellung von Höhenprofilen in der WindPRO-Applikation verwendet werden können.

Abbildung 30 stellt den schematischen Aufbau des Geoprozesses dar, wobei sich grundsätzlich der Interpolationsalgorithmus beim ersten Tool/Algorithmus je nach Interpolationsmethode („IDW“, „Kriging“ bzw. „Spline“) unterscheidet. Neben dem jeweils verwendeten Interpolationsalgorithmus, und den drei daraufhin folgenden (Geo-)Algorithmen, werden auch die Zwischen- bzw. Endergebnisse sowie die je nach Interpolationsverfahren notwendigen (Eingabe-)Parameter dargestellt:



Abbildung 30: Schematische Darstellung des Geoprozesses

Um die zu erstellenden Interpolationsoberflächen anschließend gegenüberstellen und beurteilen zu können, werden, bezogen auf alle drei Basisdatensätze, jeweils idente Vorgabewerte für die (Eingabe)-Parameter definiert. Diese Vorgabewerte sind, in Hinblick auf eine lückenlose Reproduzierbarkeit aller Interpolationsergebnisse in der vorliegenden Master Thesis, in den nachfolgenden Tabellen 6 bis 8 eindeutig spezifiziert und dargestellt:

<b>"IDW" - Interpolationsalgorithmus</b>	
<b>Data / Parameter</b>	<b>File / Value</b>
Source-Data	BEV-5x5m.shp / BEV-25x25m.shp / SRTM-90x90m.shp
Z Value Field	Elevation
Cell Size	1 (bzw. 10 beim SRTM-90x90m Datensatz)
Power (varies)	0,25 / 0,5 / 0,75 / 1 / 1,25 / 1,5 / 1,75 / 2 / 2,25 / 2,5 / 2,75 / 3
Search Radius	VARIABLE 12
Target-Data	ZE1IDW.tif

Tabelle 6: Vorgabewerte – „IDW“-Interpolationsalgorithmus

<b>"KRIGING" - Interpolationsalgorithmus</b>	
<b>Data / Parameter</b>	<b>File / Value</b>
Source-Data	BEV-5x5m.shp / BEV-25x25m.shp / SRTM-90x90m.shp
Z Value Field	Elevation
Semivariogram Properties (varies)	Spherical
	Exponential
	Gaussian
	Linear Drift
Output Cell Size	1 (bzw. 10 beim SRTM-90x90m Datensatz)
Search Radius	VARIABLE 12
Target-Data	ZE1Kriging.tif

Tabelle 7: Vorgabewerte – „Kriging“-Interpolationsalgorithmus

<b>"SPLINE" - Interpolationsalgorithmus</b>	
<b>Data / Parameter</b>	<b>File / Value</b>
Source-Data	BEV-5x5m.shp / BEV-25x25m.shp / SRTM-90x90m.shp
Z Value Field	Elevation
Output Cell Size	1 (bzw. 10 beim SRTM-90x90m Datensatz)
Spline Type	REGULARIZED
Weight (varies)	0,05 / 0,1 / 0,15 / 0,2 / 0,25 / 0,3 / 0,35 / 0,4 / 0,45 / 0,5
Number of Points	12
Target-Data	ZE1Spline.tif

Tabelle 8: Vorgabewerte – „Spline“-Interpolationsalgorithmus

## 4.2 Sensitivitätsanalyse

Um die Sensitivität bzw. Empfindlichkeit der Interpolationsalgorithmen zu prüfen wird beim „IDW“- und beim „Spline“-Verfahren jeweils ein vordefinierter Parameterraum verwendet.

- Beim „IDW“-Interpolationsalgorithmus variiert hierfür der „Power“-Parameter, welcher standardmäßig bei ESRI ArcGIS Pro Werte zwischen 0 und 3 annehmen kann.  
Aufteilung der Werte → siehe Tabelle 6 – Vorgabewerte „IDW“-Interpolation
- Beim „Spline“-Interpolationsalgorithmus variiert hierfür der „Weight“-Parameter, welcher standardmäßig bei ESRI ArcGIS Pro Werte zwischen 0 und 0,5 annehmen kann.  
Aufteilung der Werte → siehe Tabelle 8 – Vorgabewerte „Spline“-Interpolation

Prinzipiell wird bei beiden Interpolationsverfahren jeweils der gesamte zur Verfügung stehende Parameterbereich verwendet und damit geprüft ob bzw. wie sensitiv diese beiden Interpolationsmethoden auf die Eingabe unterschiedlicher Werte, bezogen auf deren Haupteingabeparametern reagieren. Die Ergebnisse der Sensitivitätsanalyse sind im Kapitel 5.3 – Kreuzvalidierung ersichtlich und dokumentiert, wobei diese wiederum für jeden der drei Basisdatensätze separat erstellt wurden.

## 4.3 Erstellung des Python Add-Ins

Der Geoprozess und somit die benutzerspezifische Erweiterung in Form eines Python Add-Ins, mit dessen Hilfe die zu vergleichenden Interpolationsoberflächen berechnet und erstellt werden, wird mit Hilfe der in ESRI ArcGIS Pro implementierten Entwicklungsumgebung erstellt. Somit wird das Python Add-In anhand des Python-Fensters direkt in der GIS-Applikation entworfen. Mit Hilfe des Python-Fensters können grundsätzlich alle programmspezifischen „*Geoprocessing-Tools*“ und deren zugrunde liegenden (Geo-)Algorithmen, Python Add-Ins sowie die gesamte Python-Funktionalität unmittelbar aus ArcGIS heraus verwendet werden (ULFERTS 2017).

Nach der Fertigstellung wird der Geoprozess als eigenständige Datei gespeichert:

- ArcGIS Pro Python Add-In → **ArcGIS-Pro\_Python\_Add-In.py**

Grundsätzlich ist es möglich Python Add-Ins auch mit Hilfe von externen Programmen bzw. Entwicklungsplattformen zu erstellen. Hierzu können Softwareprodukte wie „PythonWin“, „PyScripter“ oder auch „PyCharm“ verwendet werden.

Mit diesen Applikationen können alle, in der jeweils verwendeten GIS-Applikation, zur Verfügung stehenden Objekte, Klassen, Funktionen etc. genutzt werden. Weiters wird meist auch „*Syntax-Highlighting*“ und „*Autocompletion*“ unterstützt.

Zum nachträglichen Editieren von Python Add-Ins reicht letztlich bereits der betriebssysteminterne Editor aus um z.B. Änderungen an den erstellten Programmen vorzunehmen (TATEOSIAN 2015).

Der gesamte Geoprozess, und somit die benutzerspezifische Erweiterung in Form eines Python Add-Ins für ESRI's ArcGIS Pro, und dessen vollständiger Quellcode sind im Anhang A der vorliegenden Arbeit enthalten. Somit können alle, im Zusammenhang mit dieser Master Thesis erstellten, Interpolationsoberflächen und Ergebnisse jederzeit lückenlos reproduziert werden.

## 4.4 Implementierung

Für die Implementierung des erstellten Python Add-Ins in die GIS-Applikation stehen grundsätzlich mehrere Möglichkeiten zur Verfügung. Einerseits kann der Geoprozess direkt in der Entwicklungsumgebung, also im Python-Fenster von ESRI's ArcGIS Pro, gestartet und somit unmittelbar nach Erstellung ausgeführt werden, andererseits außerhalb der GIS-Applikation mit Hilfe der Eingabeaufforderung unter Microsoft Windows. Vorteil dieser Methode ist, dass hierfür die GIS-Applikation nicht explizit gestartet werden muss. Das Starten und Ausführen des Python Add-Ins erfolgt anhand einer einzelnen Befehlszeile, wobei diese auch in ein hierfür erstelltes „*BATCH-File*“ integriert werden kann. Für ESRI's ArcGIS Pro würde die notwendige Befehlszeile wie folgt lauten:

```
"C:\Program Files\ArcGIS\Pro\bin\Python\Scripts\propy.bat" ArcGIS_Pro_Python_Add-In.py
```

## 5. Interpolationsergebnisse

### 5.1 Visuelle Darstellung der Ergebnisse

Um einen Eindruck über die, mit Hilfe des Geoprozesses, erstellten Interpolationsoberflächen zu erhalten erfolgt im ersten Schritt eine visuelle Darstellung der Ergebnisse. Dafür werden die Interpolationsergebnisse der drei unterschiedlichen Interpolationsalgorithmen („*IDW*“, „*Kriging*“ bzw. „*Spline*“) gegenübergestellt, wobei diese je nach verwendeten Basisdatensatz 1 bis 3 einzeln dargestellt werden. Mit Hilfe einer visuellen Analyse können abweichende Muster und Unterschiede in den Ergebnissen rasch erfasst werden und die unter Umständen auftretenden Differenzen für die weiterführenden (geostatistischen) Analyseverfahren gekennzeichnet werden. Es wird dabei grundsätzlich jeweils das beste Resultat lt. der in Kapitel 5.3 durchgeführten Kreuzvalidierung je Interpolationsalgorithmus und verwendeten Basisdatensatz dargestellt.

## 5.1.1 Ergebnisse – Basisdatensatz 1

- „Inverse Distance Weighted“ – Interpolationsoberfläche mit „Power“-Parameter = 2

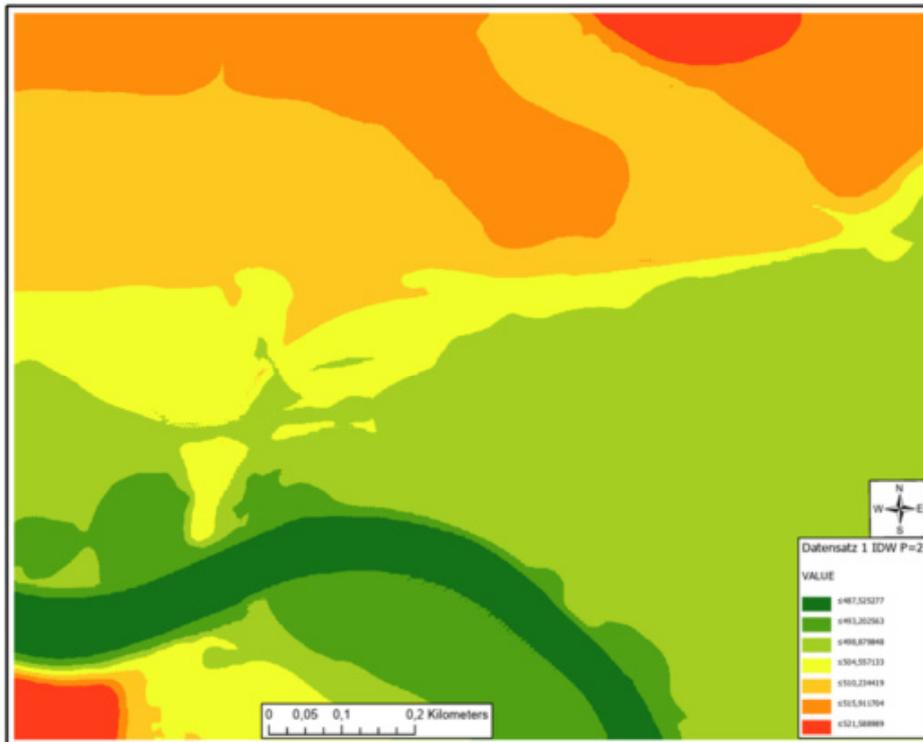


Abbildung 31: Interpolationsergebnis – „IDW“ („Power“ = 2) – Basisdatensatz 1

- „Ordinary Kriging“ – Interpolationsoberfläche mit Semivariogramm = „Exponential“

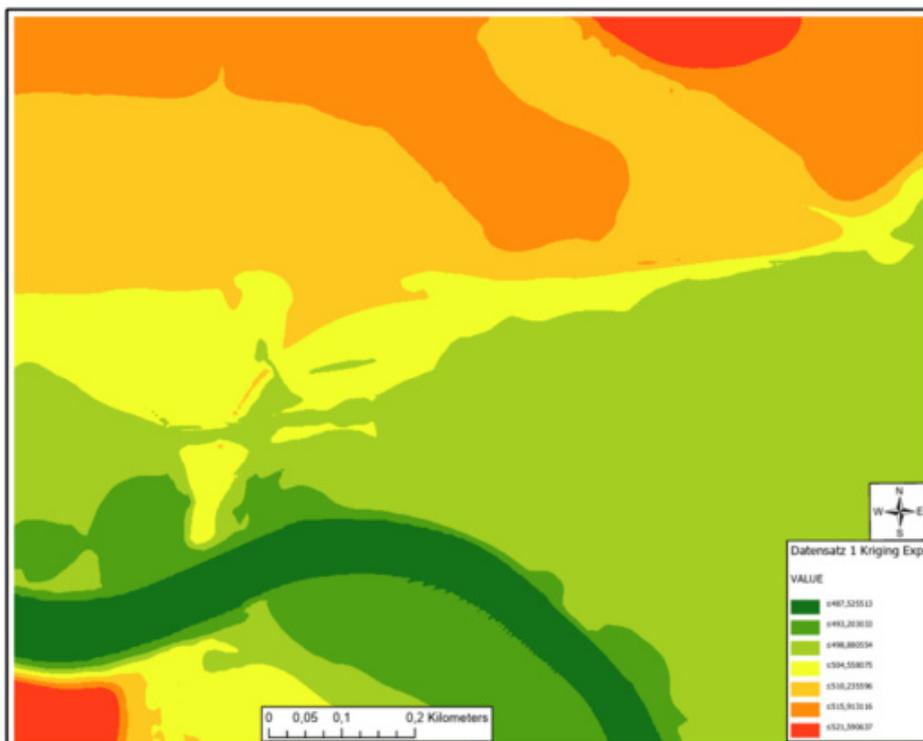


Abbildung 32: Interpolationsergebnis – „Ordinary-Kriging“ („Expon.“) – Basisdatensatz 1

- „Spline“ – Interpolationsoberfläche mit „Weight“-Parameter = 0,1

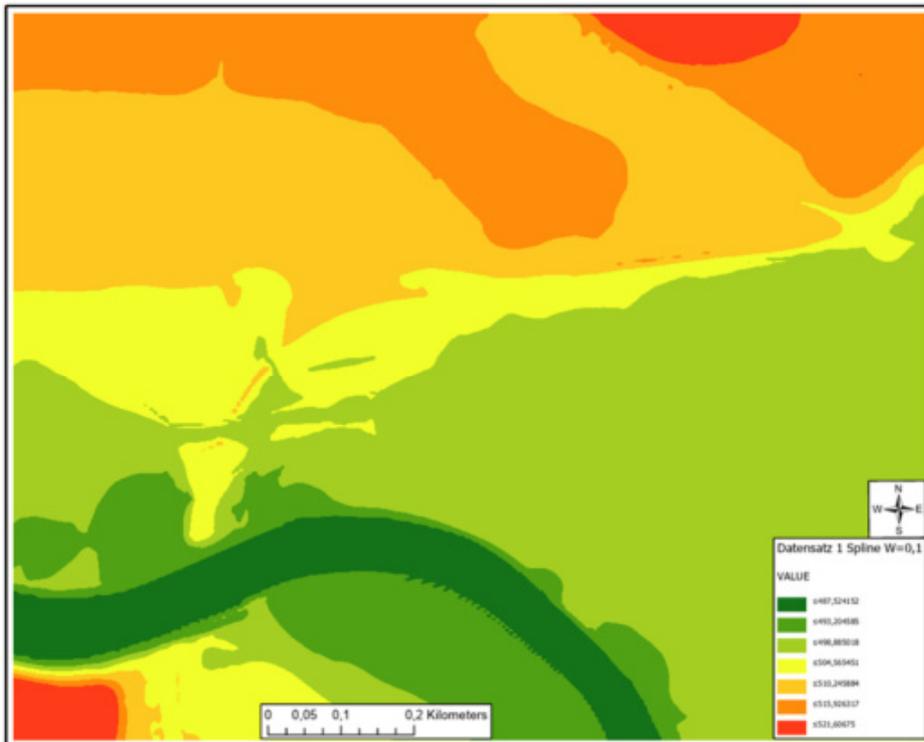


Abbildung 33: Interpolationsergebnis – „Spline“ („Weight“ = 0,1) – Basisdatensatz 1

### 5.1.2 Ergebnisse – Basisdatensatz 2

- „Inverse Distance Weighted“ – Interpolationsoberfläche mit „Power“-Parameter = 3

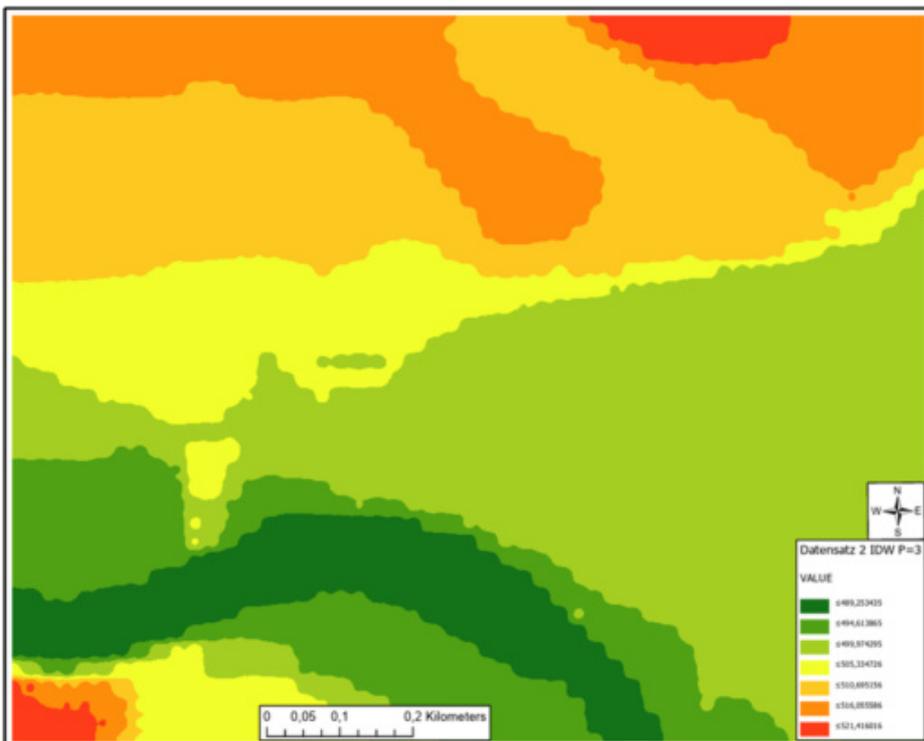


Abbildung 34: Interpolationsergebnis – „IDW“ („Power“ = 3) – Basisdatensatz 2

- „*Ordinary Kriging*“ – Interpolationsoberfläche mit Semivariogramm = „*Exponential*“

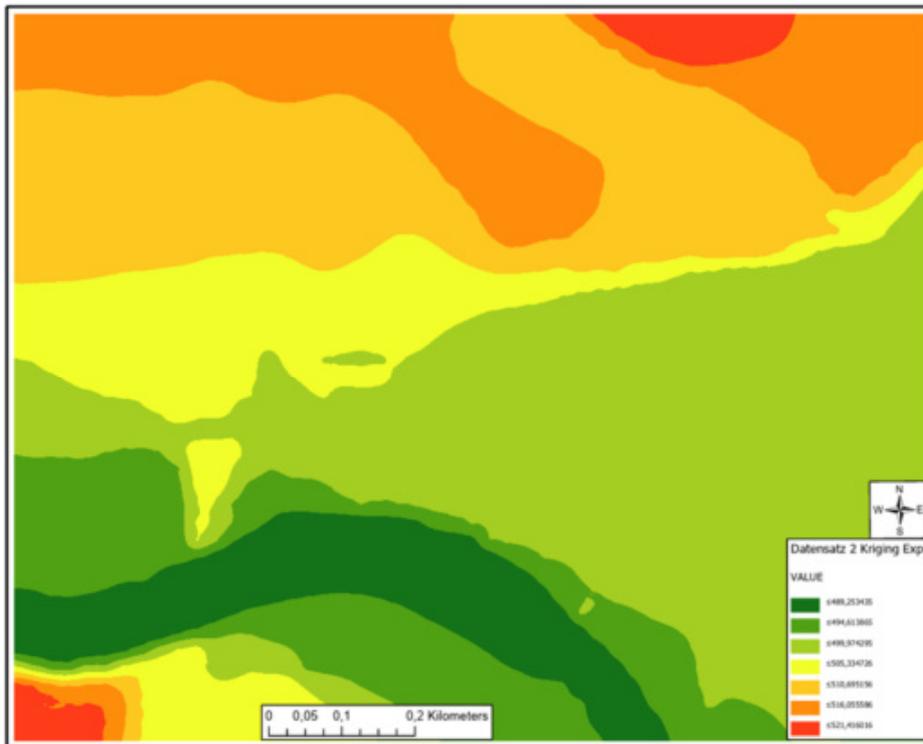


Abbildung 35: Interpolationsergebnis – „*Ordinary-Kriging*“ („*Expon.*“) – Basisdatensatz 2

- „*Spline*“ – Interpolationsoberfläche mit „*Weight*“-Parameter = 0,05

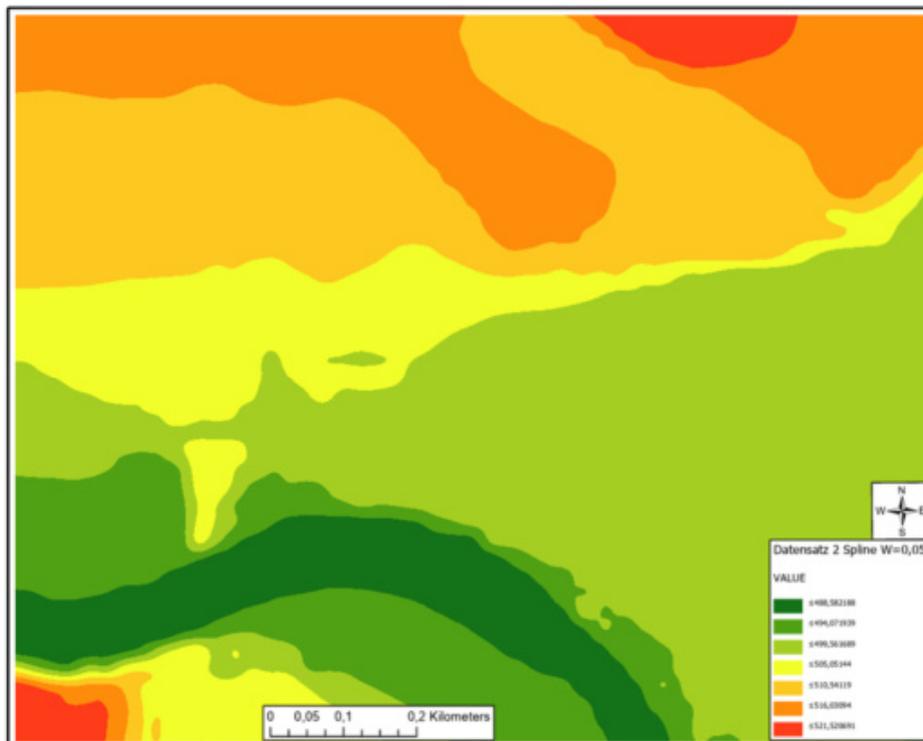


Abbildung 36: Interpolationsergebnis – „*Spline*“ („*Weight*“ = 0,05) – Basisdatensatz 2

## 5.1.3 Ergebnisse – Basisdatensatz 3

- „Inverse Distance Weighted“ – Interpolationsoberfläche mit „Power“-Parameter = 3

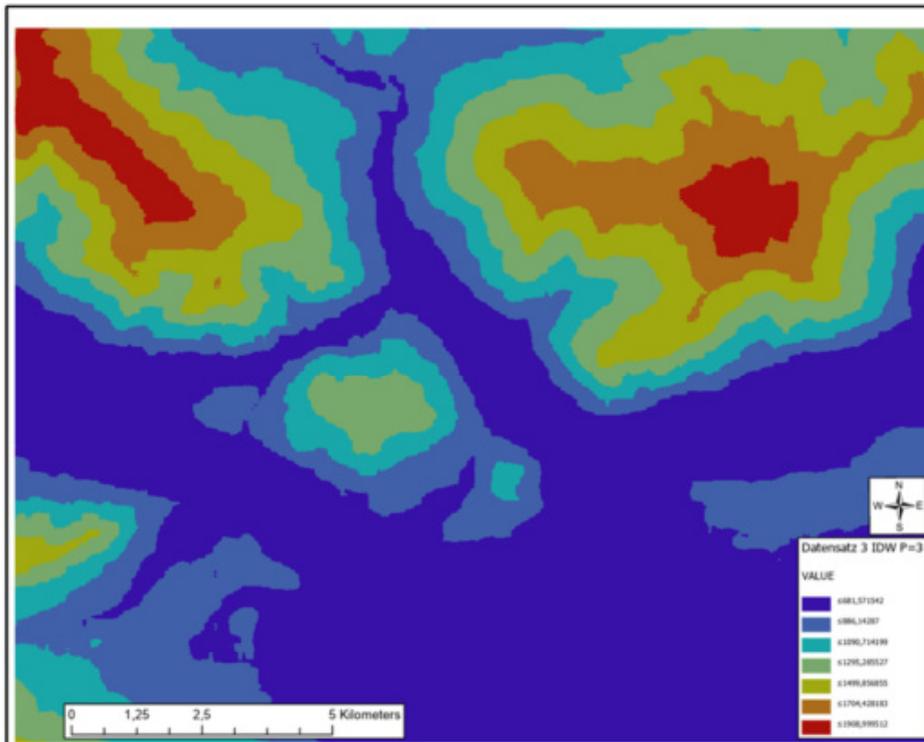


Abbildung 37: Interpolationsergebnis – „IDW“ („Power“ = 3) – Basisdatensatz 3

- „Ordinary Kriging“ – Interpolationsoberfläche mit Semivariogramm = „Exponential“

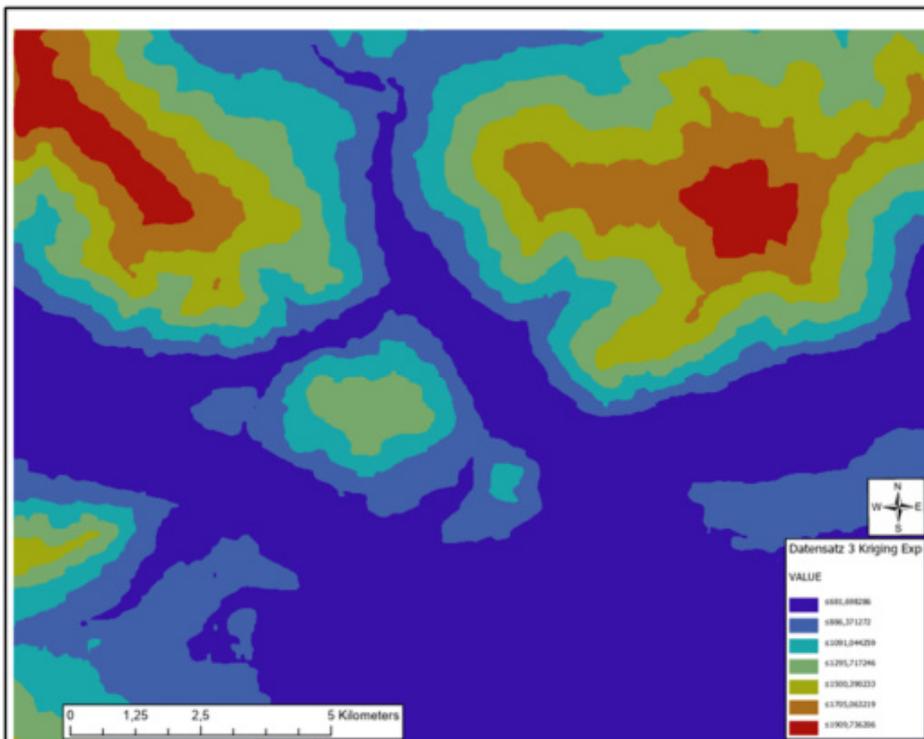


Abbildung 38: Interpolationsergebnis – „Ordinary-Kriging“ („Expon.“) – Basisdatensatz 3

- „Spline“ – Interpolationsoberfläche mit „Weight“-Parameter = 0,5

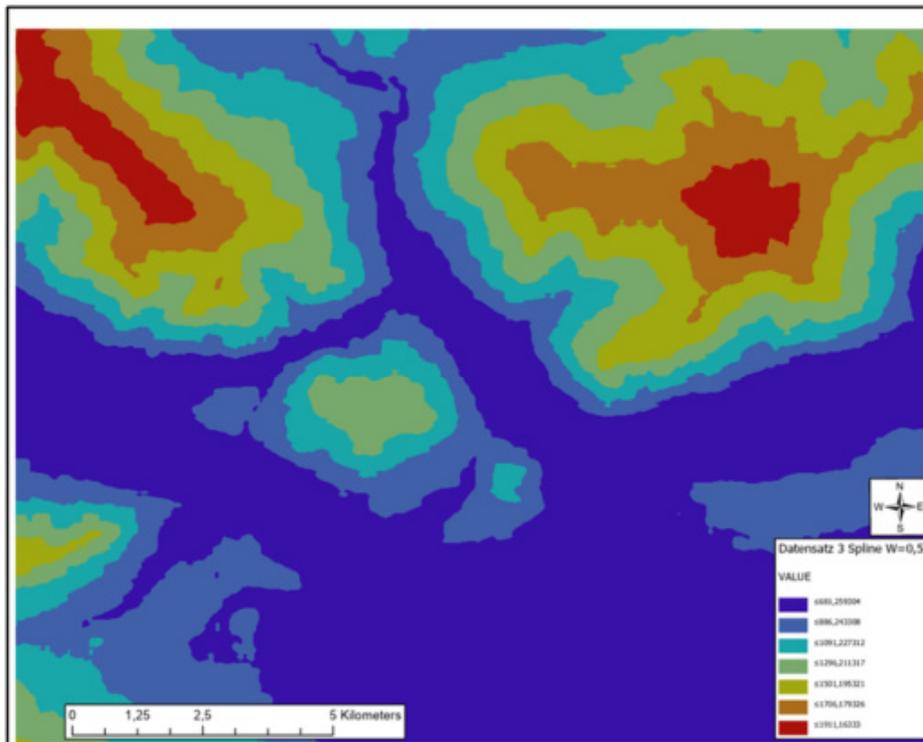


Abbildung 39: Interpolationsergebnis – „Spline“ („Weight“ = 0,5) – Basisdatensatz 3

## 5.2 Explorative Räumliche Datenanalyse (ESDA)

Nach der visuellen Darstellung der Ergebnisse erfolgt nun im zweiten Schritt eine explorative räumliche Datenanalyse („*Explorative Spatial Data Analysis*“ kurz „*ESDA*“) der berechneten Interpolationsflächen. Dabei werden die Ergebnisse der einzelnen Interpolationsverfahren systematisch untersucht, jeweils durch graphische Darstellungen charakterisiert und mittels spezifischer Kennzahlen beschrieben. Somit setzt sich der nun folgende Abschnitt primär aus zwei Hauptbestandteilen zusammen. Einerseits werden die Häufigkeitsverteilungen mit Hilfe von Histogrammen dargestellt und andererseits die Eigenschaften der einzelnen Interpolationsergebnisse anhand ihrer deskriptiven statistischen Maße beschrieben (BECKER et al. 2016).

Die explorative räumliche Datenanalyse erfolgt anhand folgender Kriterien bzw. Methoden:

- **Häufigkeitsverteilungen**

Zur Visualisierung der Häufigkeitsverteilungen werden Histogramme der Interpolationsergebnisse erstellt, wobei jeweils idente Intervalleinteilungen verwendet werden. Die Beschreibung der Lage und der Struktur der Häufigkeitsverteilungen bezüglich der Resultate steht dabei im Vordergrund (BOURIER 2018). Neben der Werteverteilung selbst wird jeweils auch der arithmetische Mittelwert (rote vertikale Gerade), der Median (lila vertikale Gerade) und die Standardabweichung (goldene vertikale Gerade) je Histogramm dargestellt. Des Weiteren ist die Normalverteilung (schwarze Kurve) ebenfalls abgebildet. Die Histogramme sind dabei im Anhang B der vorliegenden Arbeit enthalten.

- **Deskriptive statistische Maße**

Grundsätzlich wird mit Hilfe der deskriptiven Statistik ein Untersuchungsobjekt bzw. ein Untersuchungsergebnis näher beschrieben. Die Analyse besteht dabei im Ermitteln von wesentlichen Eigenschaften dieses Objektes, wobei hierfür statistische Kennzahlen wie z.B. Mittelwerte etc. berechnet werden (BOURIER 2018). Bei diesen Kennzahlen handelt es sich vorrangig um Lage-, Streuungs- bzw. Gestaltparameter, welche im weiteren Verlauf dieses Kapitels, zur systematischen Beschreibung der Resultate der einzelnen Interpolationsergebnisse verwendet werden.

Folgende statistischen Kennzahlen werden je Interpolationsergebnis ermittelt:

- |                        |                                |
|------------------------|--------------------------------|
| ○ „ <i>Minimum</i> “   | → das Minimum aller Werte      |
| ○ „ <i>Maximum</i> “   | → das Maximum aller Werte      |
| ○ „ <i>Mean</i> “      | → der arithmetische Mittelwert |
| ○ „ <i>STD.-Dev.</i> “ | → die Standardabweichung       |

Variante	Methode	Datensatz	Minimum	Maximum	Mean	STD-Dev.
D1 IDW P0,25	IDW	1	481,85	521,59	501,03	8,13
D1 IDW P0,50	IDW	1	481,85	521,59	501,03	8,13
D1 IDW P0,75	IDW	1	481,85	521,59	501,03	8,13
D1 IDW P1,00	IDW	1	481,85	521,59	501,03	8,13
D1 IDW P1,25	IDW	1	481,85	521,59	501,03	8,13
D1 IDW P1,50	IDW	1	481,85	521,59	501,03	8,13
D1 IDW P1,75	IDW	1	481,85	521,59	501,03	8,13
D1 IDW P2,00	IDW	1	481,85	521,59	501,03	8,14
D1 IDW P2,25	IDW	1	481,85	521,59	501,03	8,14
D1 IDW P2,50	IDW	1	481,85	521,59	501,03	8,14
D1 IDW P2,75	IDW	1	481,85	521,59	501,03	8,14
D1 IDW P3,00	IDW	1	481,85	521,59	501,03	8,14
D2 IDW P0,25	IDW	2	483,89	521,42	500,95	7,93
D2 IDW P0,50	IDW	2	483,89	521,42	500,95	7,94
D2 IDW P0,75	IDW	2	483,89	521,42	500,95	7,96
D2 IDW P1,00	IDW	2	483,89	521,42	500,95	7,98
D2 IDW P1,25	IDW	2	483,89	521,42	500,95	8,00
D2 IDW P1,50	IDW	2	483,89	521,42	500,95	8,02
D2 IDW P1,75	IDW	2	483,89	521,42	500,95	8,04
D2 IDW P2,00	IDW	2	483,89	521,42	500,95	8,05
D2 IDW P2,25	IDW	2	483,89	521,42	500,95	8,07
D2 IDW P2,50	IDW	2	483,89	521,42	500,95	8,08
D2 IDW P2,75	IDW	2	483,89	521,42	500,95	8,09
D2 IDW P3,00	IDW	2	483,89	521,42	500,95	8,10
D3 IDW P0,25	IDW	3	483,33	1902,02	901,10	374,93
D3 IDW P0,50	IDW	3	479,67	1904,00	901,10	375,00
D3 IDW P0,75	IDW	3	478,28	1906,18	901,10	375,08
D3 IDW P1,00	IDW	3	478,00	1907,72	901,10	375,16
D3 IDW P1,25	IDW	3	477,36	1908,49	901,10	375,24
D3 IDW P1,50	IDW	3	477,13	1908,81	901,10	375,32
D3 IDW P1,75	IDW	3	477,04	1908,93	901,10	375,40
D3 IDW P2,00	IDW	3	477,01	1908,97	901,10	375,47
D3 IDW P2,25	IDW	3	477,00	1909,00	901,10	375,53
D3 IDW P2,50	IDW	3	477,00	1909,00	901,10	375,58
D3 IDW P2,75	IDW	3	477,00	1909,00	901,10	375,62
D3 IDW P3,00	IDW	3	477,00	1909,00	901,10	375,66
D1 SPH	Kriging	1	481,85	521,59	501,03	8,14
D1 EXP	Kriging	1	481,85	521,59	501,03	8,14
D1 GAU	Kriging	1	481,85	521,59	501,03	8,12
D1 LiDr	Kriging	1	481,85	521,59	501,04	8,13
D2 SPH	Kriging	2	483,89	521,42	500,95	8,13
D2 EXP	Kriging	2	483,89	521,42	500,95	8,13
D2 GAU	Kriging	2	483,89	521,42	500,95	7,92
D2 LiDr	Kriging	2	483,89	521,42	500,95	7,93
D3 SPH	Kriging	3	483,33	1902,31	901,10	374,95
D3 EXP	Kriging	3	477,03	1909,74	901,10	375,77
D3 GAU	Kriging	3	483,25	1901,50	901,10	374,87
D3 LiDr	Kriging	3	478,36	1901,76	901,10	374,84

D1 SPL_W0,05	Spline	1	481,84	521,61	501,03	8,15
D1 SPL_W0,10	Spline	1	481,84	521,61	501,03	8,15
D1 SPL_W0,15	Spline	1	481,84	521,61	501,03	8,15
D1 SPL_W0,20	Spline	1	481,84	521,61	501,03	8,15
D1 SPL_W0,25	Spline	1	481,84	521,61	501,03	8,15
D1 SPL_W0,30	Spline	1	481,84	521,61	501,03	8,15
D1 SPL_W0,35	Spline	1	481,84	521,61	501,03	8,15
D1 SPL_W0,40	Spline	1	481,84	521,61	501,03	8,15
D1 SPL_W0,45	Spline	1	481,84	521,61	501,03	8,15
D1 SPL_W0,50	Spline	1	481,84	521,61	501,03	8,15
D2 SPL_W0,05	Spline	2	483,09	521,52	500,95	8,16
D2 SPL_W0,10	Spline	2	483,11	521,54	500,95	8,17
D2 SPL_W0,15	Spline	2	483,12	521,54	500,95	8,17
D2 SPL_W0,20	Spline	2	483,13	521,55	500,95	8,17
D2 SPL_W0,25	Spline	2	483,14	521,55	500,95	8,17
D2 SPL_W0,30	Spline	2	483,14	521,56	500,95	8,17
D2 SPL_W0,35	Spline	2	483,15	521,56	500,95	8,17
D2 SPL_W0,40	Spline	2	483,15	521,56	500,95	8,17
D2 SPL_W0,45	Spline	2	483,15	521,56	500,95	8,17
D2 SPL_W0,50	Spline	2	483,16	521,56	500,95	8,17
D3 SPL_W0,05	Spline	3	476,36	1910,84	901,09	375,90
D3 SPL_W0,10	Spline	3	476,31	1910,96	901,09	375,90
D3 SPL_W0,15	Spline	3	476,30	1911,03	901,09	375,90
D3 SPL_W0,20	Spline	3	476,29	1911,07	901,09	375,90
D3 SPL_W0,25	Spline	3	476,28	1911,09	901,09	375,90
D3 SPL_W0,30	Spline	3	476,28	1911,11	901,09	375,90
D3 SPL_W0,35	Spline	3	476,28	1911,13	901,09	375,90
D3 SPL_W0,40	Spline	3	476,28	1911,14	901,09	375,90
D3 SPL_W0,45	Spline	3	476,28	1911,15	901,09	375,90
D3 SPL_W0,50	Spline	3	476,28	1911,16	901,09	375,90

Tabelle 9: Deskriptive statistische Maße aller erstellten Interpolationsoberflächen

### 5.3 Kreuzvalidierung

Die Kreuzvalidierung (im englisch sprachigen Raum als „*Cross-Validation*“ bezeichnet) wird grundsätzlich zur Überprüfung der Modellgüte von Interpolationsoberflächen verwendet und dient somit im weiteren Verlauf zur Haupt-Validierung der Ergebnisse der mit Hilfe des Geoprozesses erstellten Höhenoberflächen (EISANK & KOCH 2017c). Für die Kreuzvalidierung werden vor den durchzuführenden Interpolationsberechnungen, also vor der Erstellung der einzelnen Oberflächen mit Hilfe der drei unterschiedlichen Interpolationsverfahren „*IDW*“, „*Kriging*“ bzw. „*Spline*“ und den in Kapitel 4.1 beschriebenen Parametern, jeweils eine definierte Menge an bekannten Messpunkten (sogenannte Referenzpunkte) aus den unterschiedlichen Eingangsdatenquellen (Basisdatensatz 1 bis 3) entfernt und anschließend deren bereits bekannten (Höhen-)Werte mit den dazugehörigen berechneten Zellenwerten der Interpolationsoberflächen verglichen. In diesem Zusammenhang werden zwei spezifische Kennzahlen ermittelt anhand deren letztlich die Aussage bezüglich der jeweils am besten geeigneten Interpolationsoberfläche getroffen wird.

Die Kreuzvalidierung erfolgt auf Basis der Ermittlung folgender zwei spezifischen Kennzahlen:

- **„Mean Error“ („ME“)**

Mit Hilfe des „*Mean Errors*“ wird jeweils die mittlere Differenz zwischen den gemessenen und den vorhergesagten Werten berechnet.

Der „*Mean Error*“ wird folgendermaßen berechnet:

$$\frac{\sum_{i=1}^n (\hat{Z}(\mathbf{s}_i) - z(\mathbf{s}_i))}{n}$$

Quelle: ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2019)

- **“Root Mean Square Error” (“RMSE”)**

Der „*Root Mean Square Error*“ errechnet sich als Wurzel der Summe der Abweichungsquadrate und dient der Beurteilung wie exakt das jeweilige Interpolationsmodell bzw. die jeweils daraus berechnete Oberfläche die gemessenen Werte vorhersagt.

Der „*Root Mean Square Error*“ wird folgendermaßen berechnet:

$$\sqrt{\frac{\sum_{i=1}^n (\hat{Z}(\mathbf{s}_i) - z(\mathbf{s}_i))^2}{n}}$$

Quelle: ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2019)

Somit werden in weiterer Folge für alle erstellten Interpolationsoberflächen diese beiden Kennzahlen ermittelt und anschließend tabellarisch und graphisch gegenübergestellt, wobei dies wiederum getrennt nach Basisdatensatz 1 bis 3 geschieht. Grundsätzlich gilt → Je geringer die resultierenden Werte für den „Mean Error“ und den „Root Mean Square Error“ ausfallen desto besser erfolgt die Annäherung der jeweils berechneten Interpolationsoberfläche an die tatsächliche Situation.

Die nachfolgende Tabelle 10 spezifiziert alle Referenzpunkte und deren Koordinaten (mit „Austria GK Central“ als Koordinatensystem/Projektion) samt deren Referenz-Höhenwerten [in Meter] hinsichtlich der für den Vergleich der Interpolationsalgorithmen verwendeten Basisdatensätze 1 bis 3. Für die Basisdatensätze 1 und 2 werden aufgrund der gleichen räumlichen Ausprägung der Testdaten idente Referenzpunkte verwendet. Beim Basisdatensatz 3 werden, basierend auf den abweichenden räumlichen Bereich der „SRTM“-Daten, dementsprechend angepasste Referenzpunkte gewählt.

Basisdatensatz 1 und 2				Basisdatensatz 3			
POINT	POINT_X	POINT_Y	POINT_Z	POINT	POINT_X	POINT_Y	POINT_Z
OID	EPSG-Code: 31255			OID	EPSG-Code: 31255		
0	39950	164975	514,85	0	30965	176285	1735,00
1	38825	164925	514,10	1	47270	176128	1350,00
2	39500	164875	509,80	2	39588	175501	1037,00
3	39700	164875	508,95	3	44291	175266	1419,00
4	38900	164850	507,63	4	36374	174560	963,00
5	39150	164850	508,34	5	32612	174482	1743,00
6	39875	164775	512,19	6	40215	172757	1386,00
7	39425	164725	513,06	7	33709	171503	1223,00
8	38975	164675	505,81	8	45859	171425	1360,00
9	39675	164675	509,06	9	35120	169778	631,00
10	38875	164525	501,49	10	31044	169700	511,00
11	39275	164500	497,65	11	47270	169387	499,00
12	39625	164475	498,25	12	39039	169308	808,00
13	39050	164425	497,82	13	42880	168759	501,00
14	39300	164275	485,16	14	46486	167192	703,00
15	39075	164250	484,82	15	33239	167113	712,00
16	39925	164225	495,48	16	40137	166878	643,00
17	39675	164200	495,41	17	37158	165232	556,00
18	38850	164175	484,67	18	43194	165075	508,00
19	39275	164075	492,89	19	33552	164291	744,00
20	39550	164075	484,29	20	39353	163429	500,00
21	38800	164050	516,26	21	30887	163272	1141,00
22	39875	164050	494,71	22	47740	163116	541,00

Tabelle 10: Referenzpunkte – Basisdatensatz 1 bis 3

Die nachfolgenden Abbildungen 40 bis 43 geben einen Überblick bezüglich der räumlichen Verteilung aller Referenzpunkte, wobei diese je nach verwendeten Basisdatensatz getrennt dargestellt werden. Die ersten beiden Abbildungen zeigen somit die Referenzpunkte bezogen auf die Basisdatensätze 1 bzw. 2 (BEV-Daten) und die anschließend folgenden beiden Abbildungen stellen die Referenzpunkte für den Basisdatensatz 3 („SRTM“-Daten) dar.

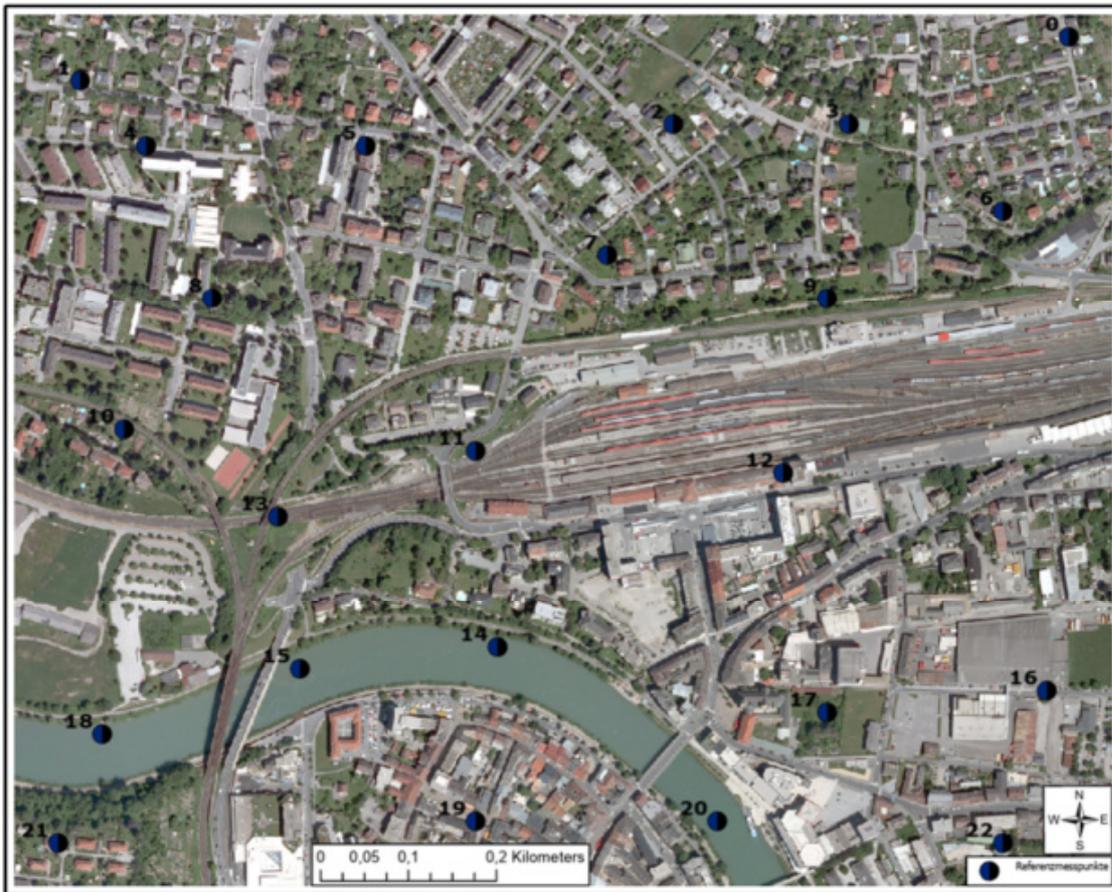


Abbildung 40: Räumliche Verteilung der Referenzpunkte → Basisdatensatz 1 und 2

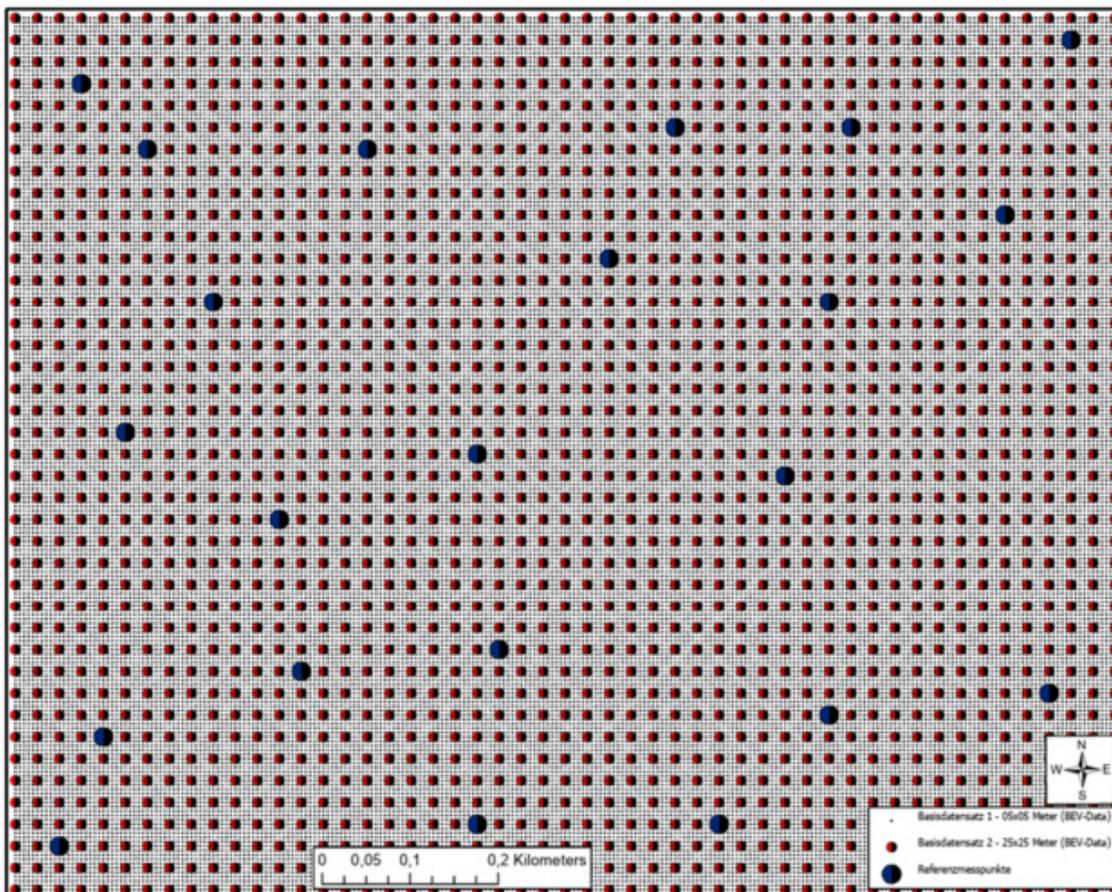


Abbildung 41: Referenzpunkte inklusive Basisdatensätze 1 und 2 („BEV“-Daten)

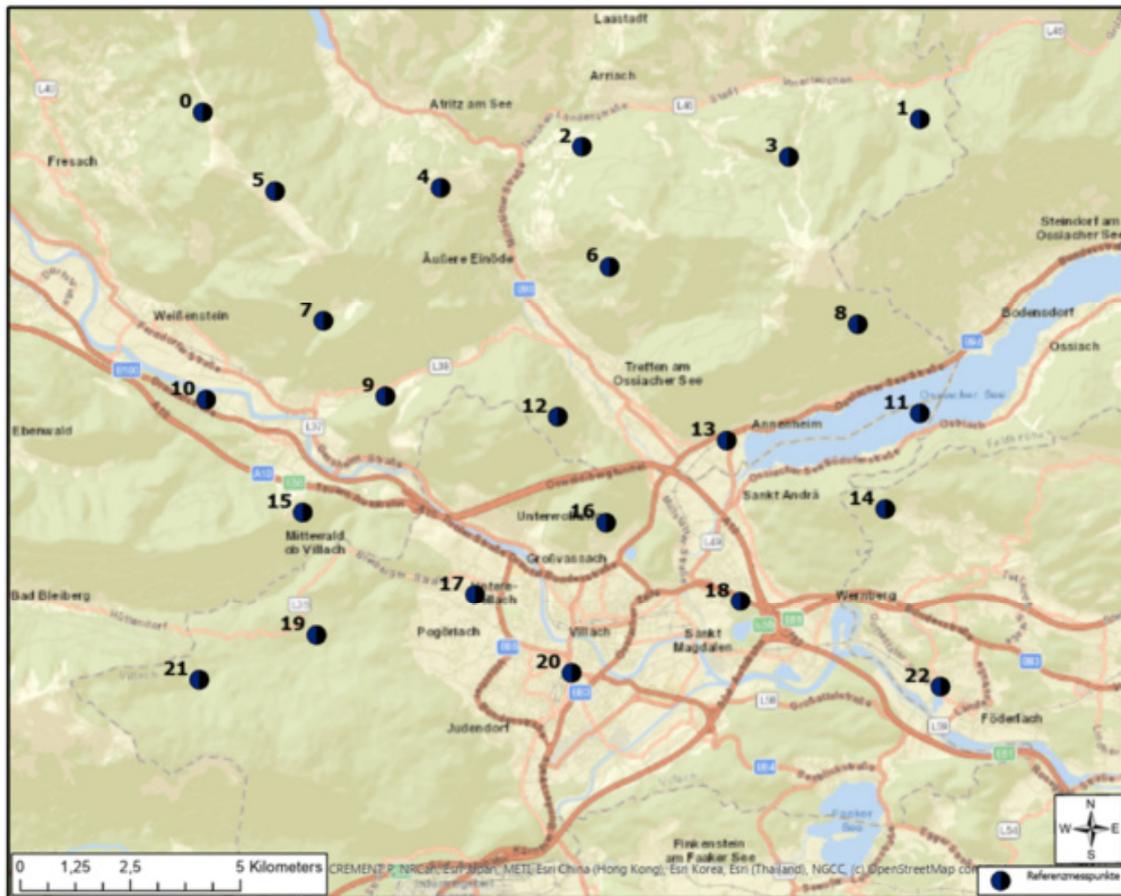


Abbildung 42: Räumliche Verteilung der Referenzpunkte → Basisdatensatz 3

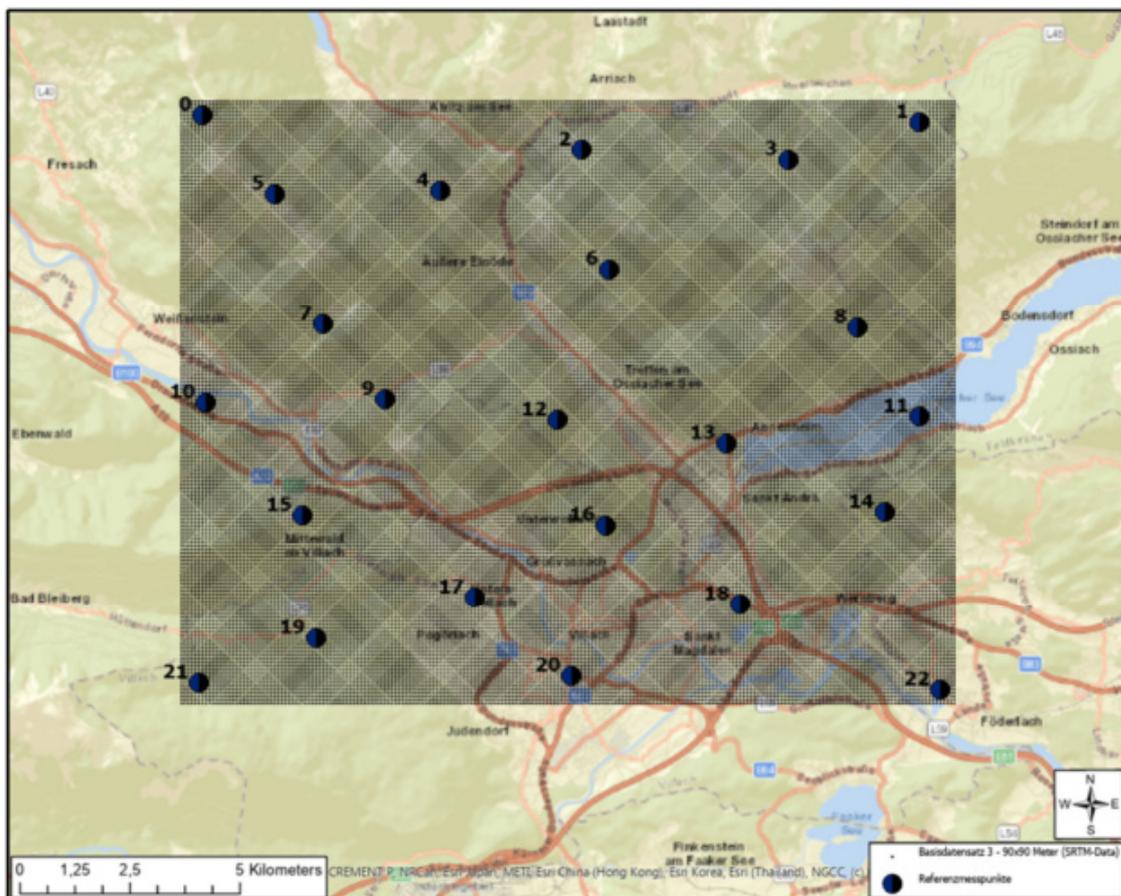


Abbildung 43: Referenzpunkte inklusive Basisdatensatz 3 („SRTM“-Daten)

## 5.3.1 Ergebnisse – Basisdatensatz 1

Die nachfolgende Tabelle 11 zeigt die, bezogen auf den Basisdatensatz 1, ermittelten Fehlerwerte „Mean Errors“ und „Root Mean Square Errors“ für alle berechneten Interpolationsoberflächen, wobei auch die Sensitivität des „IDW“-Verfahrens anhand des variablen „Power“-Parameters [Parameterraum: 0,25; 0,50; 0,75; 1,00; 1,25; 1,50; 1,75; 2,00; 2,25; 2,50; 2,75; 3,0], und beim „Spline“-Verfahren anhand eines variablen „Weight“-Parameters [Parameterraum: 0,05; 0,10; 0,15; 0,20; 0,25; 0,30; 0,35; 0,40; 0,45; 0,50] dargestellt wird.

Ergebnis	Power / Weight / Var.	Datensatz	Algorithmus	ME	RMSE
D1_IDW_P0,25	0,25	1	IDW	0,06	0,45
D1_IDW_P0,50	0,50	1	IDW	0,06	0,45
D1_IDW_P0,75	0,75	1	IDW	0,06	0,45
D1_IDW_P1,00	1,00	1	IDW	0,06	0,45
D1_IDW_P1,25	1,25	1	IDW	0,05	0,45
D1_IDW_P1,50	1,50	1	IDW	0,05	0,45
D1_IDW_P1,75	1,75	1	IDW	0,05	0,45
D1_IDW_P2,00	2,00	1	IDW	0,05	0,45
D1_IDW_P2,25	2,25	1	IDW	0,05	0,45
D1_IDW_P2,50	2,50	1	IDW	0,05	0,45
D1_IDW_P2,75	2,75	1	IDW	0,05	0,45
D1_IDW_P3,00	3,00	1	IDW	0,05	0,45
D1_KRI_Spherical	Spherical	1	Kriging	0,04	0,44
D1_KRI_Exponential	Exponential	1	Kriging	0,04	0,44
D1_KRI_Gaussian	Gaussian	1	Kriging	0,06	0,45
D1_KRI_LinearDrift	LinearDrift	1	Kriging	0,06	0,45
D1_SPL_W0,05	0,05	1	Spline	0,02	0,44
D1_SPL_W0,10	0,10	1	Spline	0,02	0,44
D1_SPL_W0,15	0,15	1	Spline	0,02	0,44
D1_SPL_W0,20	0,20	1	Spline	0,02	0,44
D1_SPL_W0,25	0,25	1	Spline	0,02	0,44
D1_SPL_W0,30	0,30	1	Spline	0,02	0,44
D1_SPL_W0,35	0,35	1	Spline	0,02	0,44
D1_SPL_W0,40	0,40	1	Spline	0,02	0,44
D1_SPL_W0,45	0,45	1	Spline	0,02	0,44
D1_SPL_W0,50	0,50	1	Spline	0,02	0,44

Tabelle 11: Ergebnisse der Kreuzvalidierung („ME“ und „RMSE“) – Basisdatensatz 1

Die Abbildungen 44 und 45 stellen die, bezogen auf den Basisdatensatz 1, ermittelten Fehlerwerte anhand von Liniendiagrammen dar, wobei jede Fehlerkennzahl separat als eigenes Diagramm dargestellt wird. Weiters wird anhand der beiden Abbildungen 46 bis 47 jeweils die Sensitivität des „IDW“- bzw. „Spline“-Verfahrens gezeigt und anhand der resultierenden „RMSEs“ visualisiert.

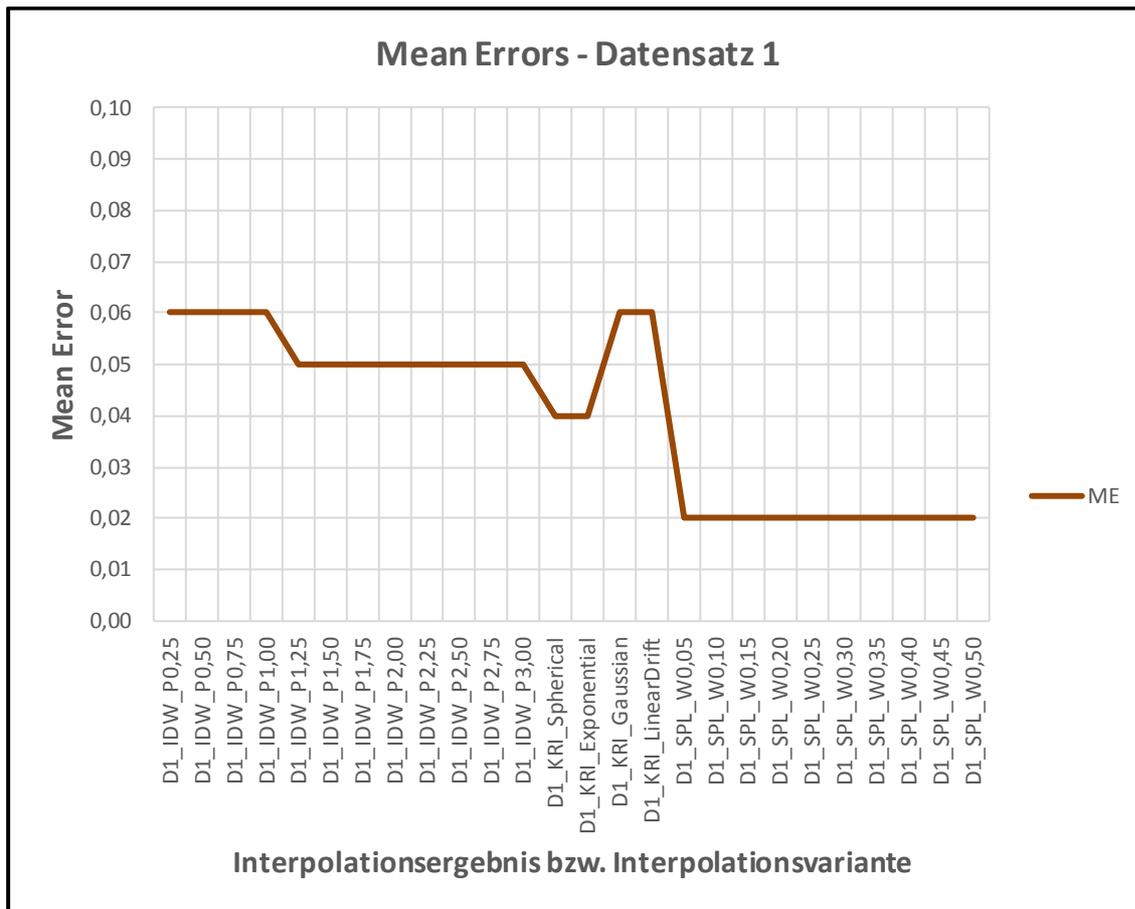


Abbildung 44: ermittelte „Mean Errors“ – Basisdatensatz 1

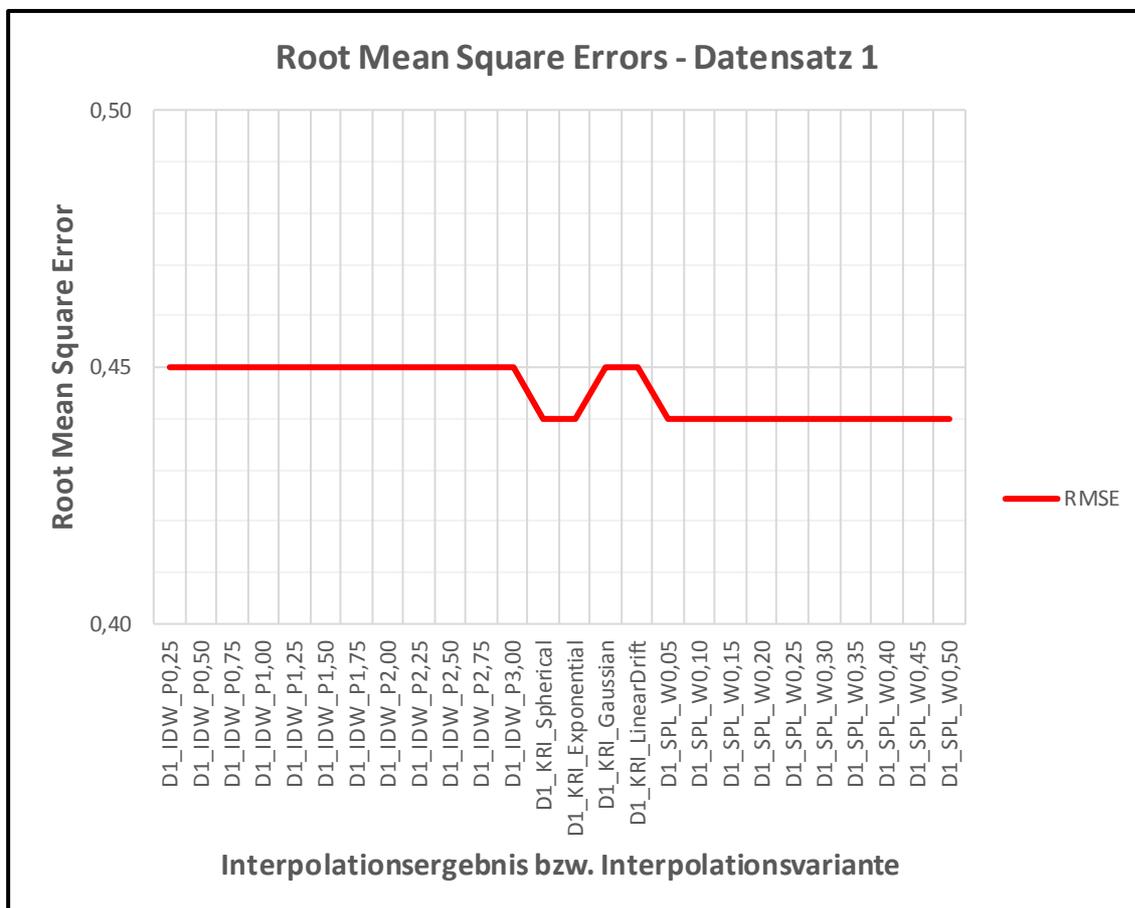


Abbildung 45: ermittelte „Root Mean Square Errors“ – Basisdatensatz 1

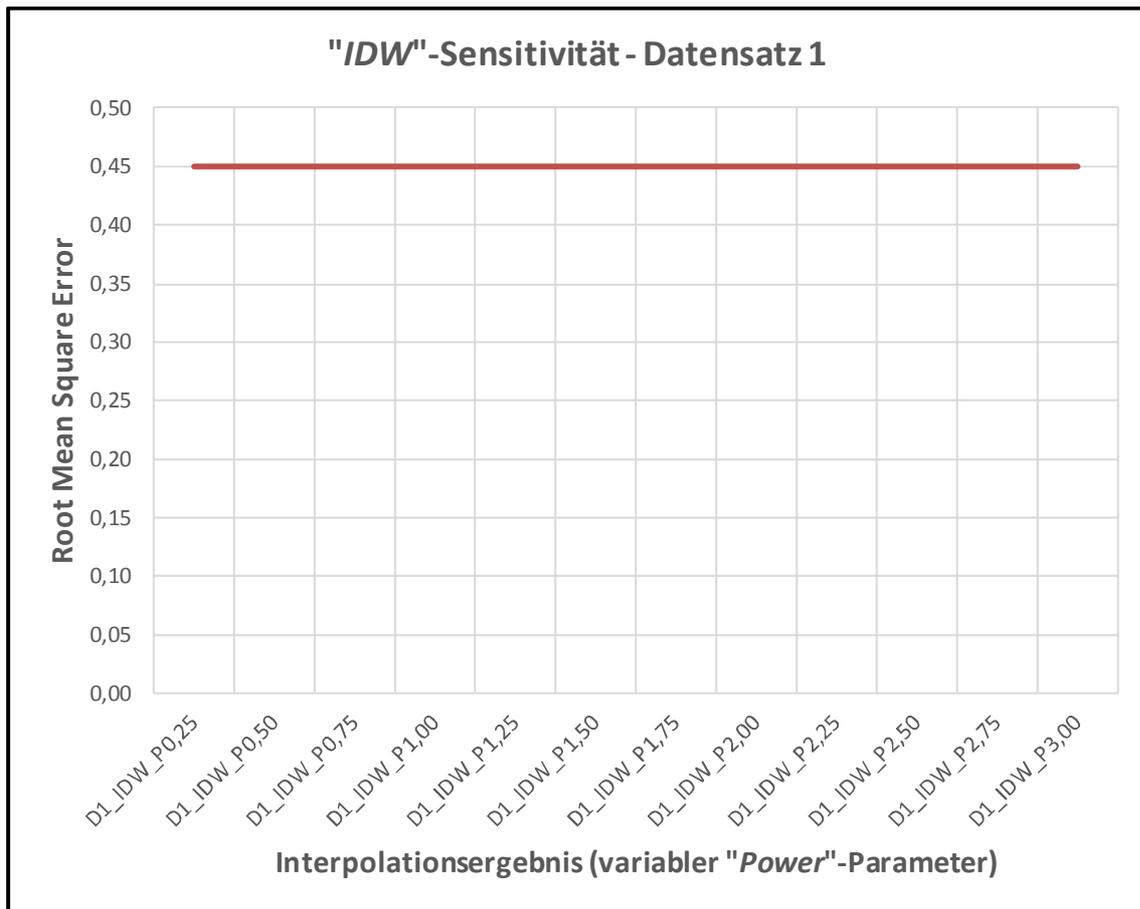


Abbildung 46: „IDW“-Sensitivität – Basisdatensatz 1

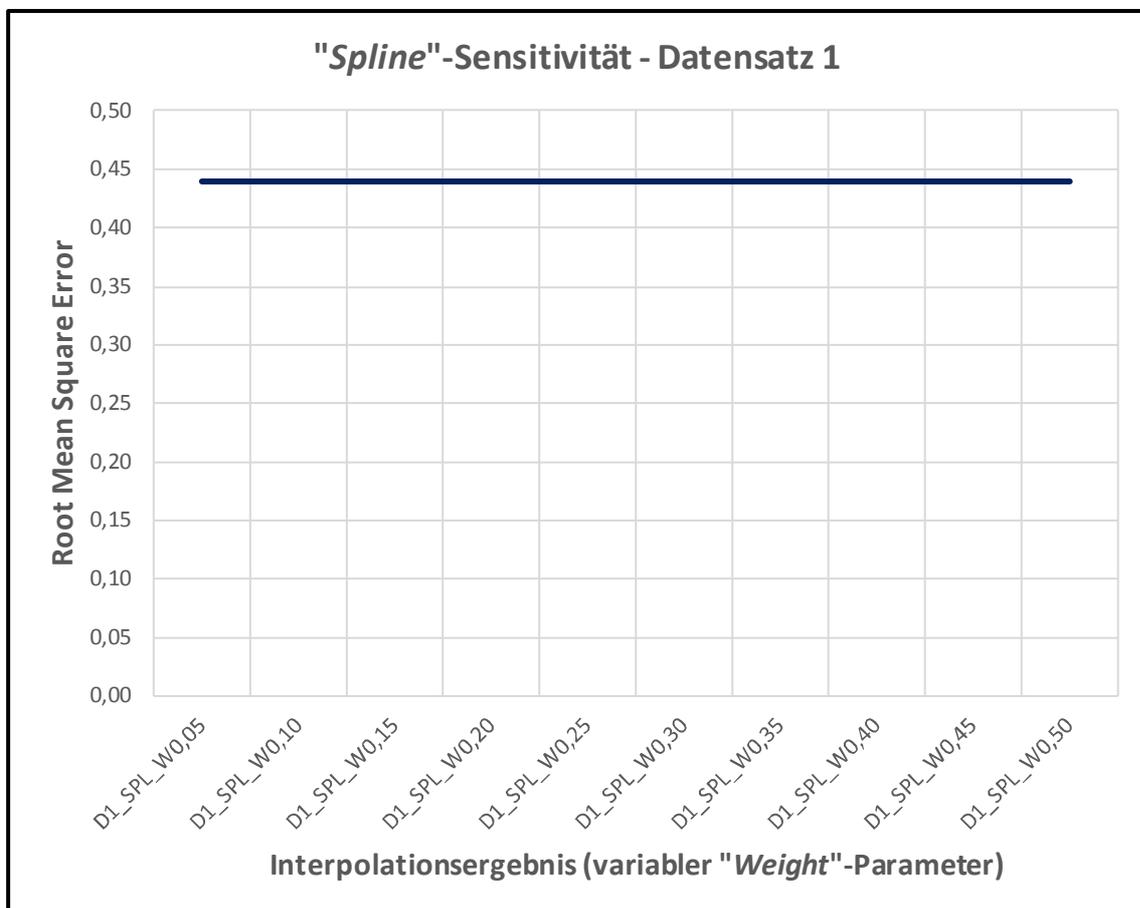


Abbildung 47: „Spline“-Sensitivität – Basisdatensatz 1

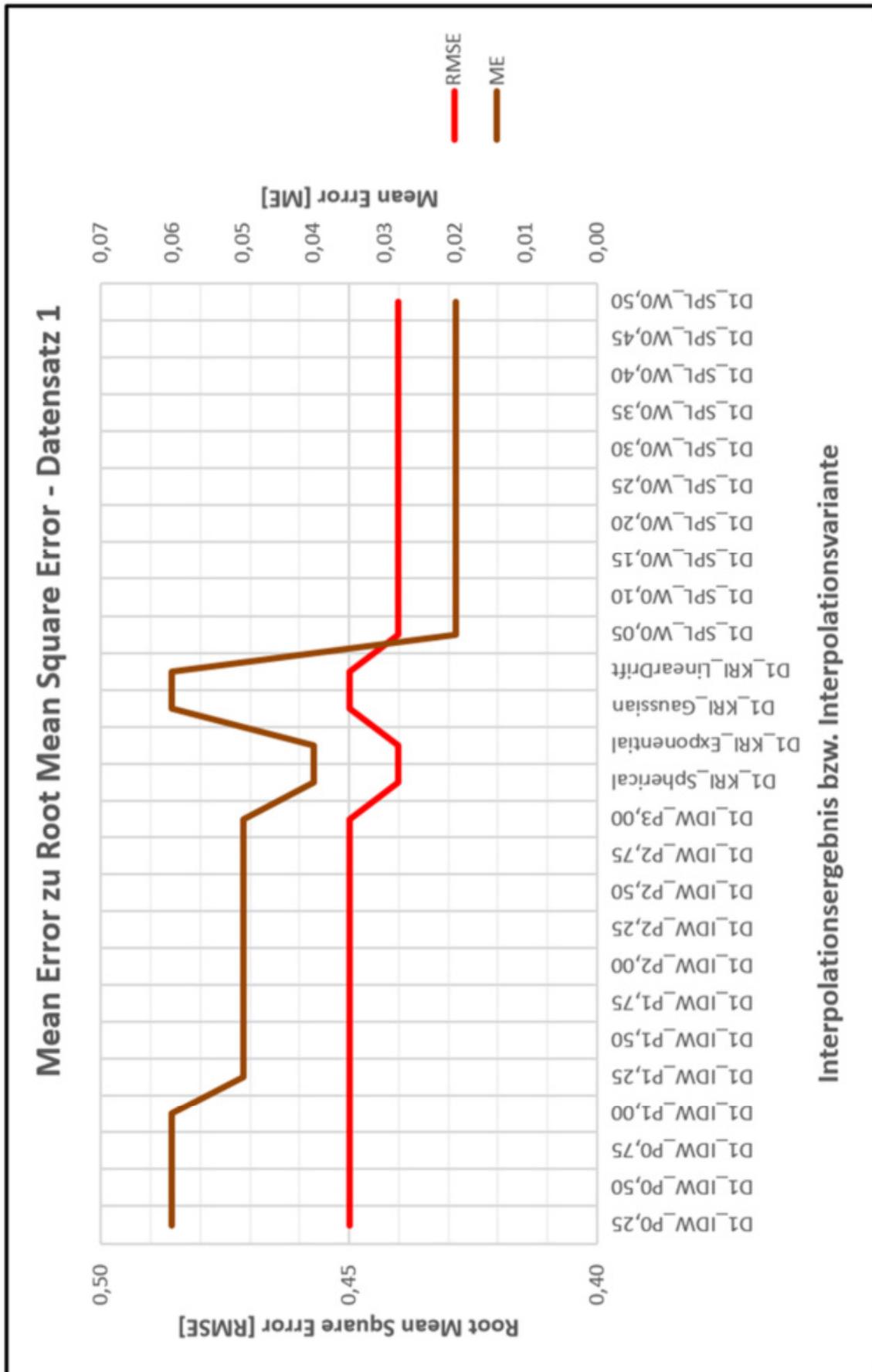


Abbildung 48: „Mean Error“ zu „Root Mean Square Error“ – Basisdatensatz 1

## 5.3.2 Ergebnisse – Basisdatensatz 2

Die nachfolgende Tabelle 12 zeigt die, bezogen auf den Basisdatensatz 2, ermittelten Fehlerwerte „Mean Errors“ und „Root Mean Square Errors“ für alle berechneten Interpolationsoberflächen, wobei auch die Sensitivität des „IDW“-Verfahrens anhand des variablen „Power“-Parameters [Parameterraum: 0,25; 0,50; 0,75; 1,00; 1,25; 1,50; 1,75; 2,00; 2,25; 2,50; 2,75; 3,0], und beim „Spline“-Verfahren anhand eines variablen „Weight“-Parameters [Parameterraum: 0,05; 0,10; 0,15; 0,20; 0,25; 0,30; 0,35; 0,40; 0,45; 0,50] dargestellt wird.

Ergebnis	Power / Weight / Var.	Datensatz	Algorithmus	ME	RMSE
D2_IDW_P0,25	0,25	2	IDW	0,17	1,28
D2_IDW_P0,50	0,50	2	IDW	0,16	1,24
D2_IDW_P0,75	0,75	2	IDW	0,16	1,21
D2_IDW_P1,00	1,00	2	IDW	0,16	1,18
D2_IDW_P1,25	1,25	2	IDW	0,16	1,14
D2_IDW_P1,50	1,50	2	IDW	0,15	1,12
D2_IDW_P1,75	1,75	2	IDW	0,15	1,09
D2_IDW_P2,00	2,00	2	IDW	0,15	1,06
D2_IDW_P2,25	2,25	2	IDW	0,14	1,04
D2_IDW_P2,50	2,50	2	IDW	0,14	1,01
D2_IDW_P2,75	2,75	2	IDW	0,14	0,99
D2_IDW_P3,00	3,00	2	IDW	0,14	0,97
D2_KRI_Spherical	Spherical	2	Kriging	0,11	0,71
D2_KRI_Exponential	Exponential	2	Kriging	0,11	0,71
D2_KRI_Gaussian	Gaussian	2	Kriging	0,16	1,31
D2_KRI_LinearDrift	LinearDrift	2	Kriging	0,16	1,31
D2_SPL_W0,05	0,05	2	Spline	0,08	0,58
D2_SPL_W0,10	0,10	2	Spline	0,09	0,59
D2_SPL_W0,15	0,15	2	Spline	0,09	0,59
D2_SPL_W0,20	0,20	2	Spline	0,09	0,59
D2_SPL_W0,25	0,25	2	Spline	0,09	0,59
D2_SPL_W0,30	0,30	2	Spline	0,09	0,60
D2_SPL_W0,35	0,35	2	Spline	0,09	0,60
D2_SPL_W0,40	0,40	2	Spline	0,09	0,60
D2_SPL_W0,45	0,45	2	Spline	0,09	0,60
D2_SPL_W0,50	0,50	2	Spline	0,09	0,60

Tabelle 12: Ergebnisse der Kreuzvalidierung („ME“ und „RMSE“) – Basisdatensatz 2

Die Abbildungen 49 und 50 stellen die, bezogen auf den Basisdatensatz 2, ermittelten Fehlerwerte anhand von Liniendiagrammen dar, wobei jede Fehlerkennzahl separat als eigenes Diagramm dargestellt wird. Weiters wird anhand der beiden Abbildungen 51 bis 52 jeweils die Sensitivität des „IDW“- bzw. „Spline“-Verfahrens gezeigt und anhand der resultierenden „RMSEs“ visualisiert.

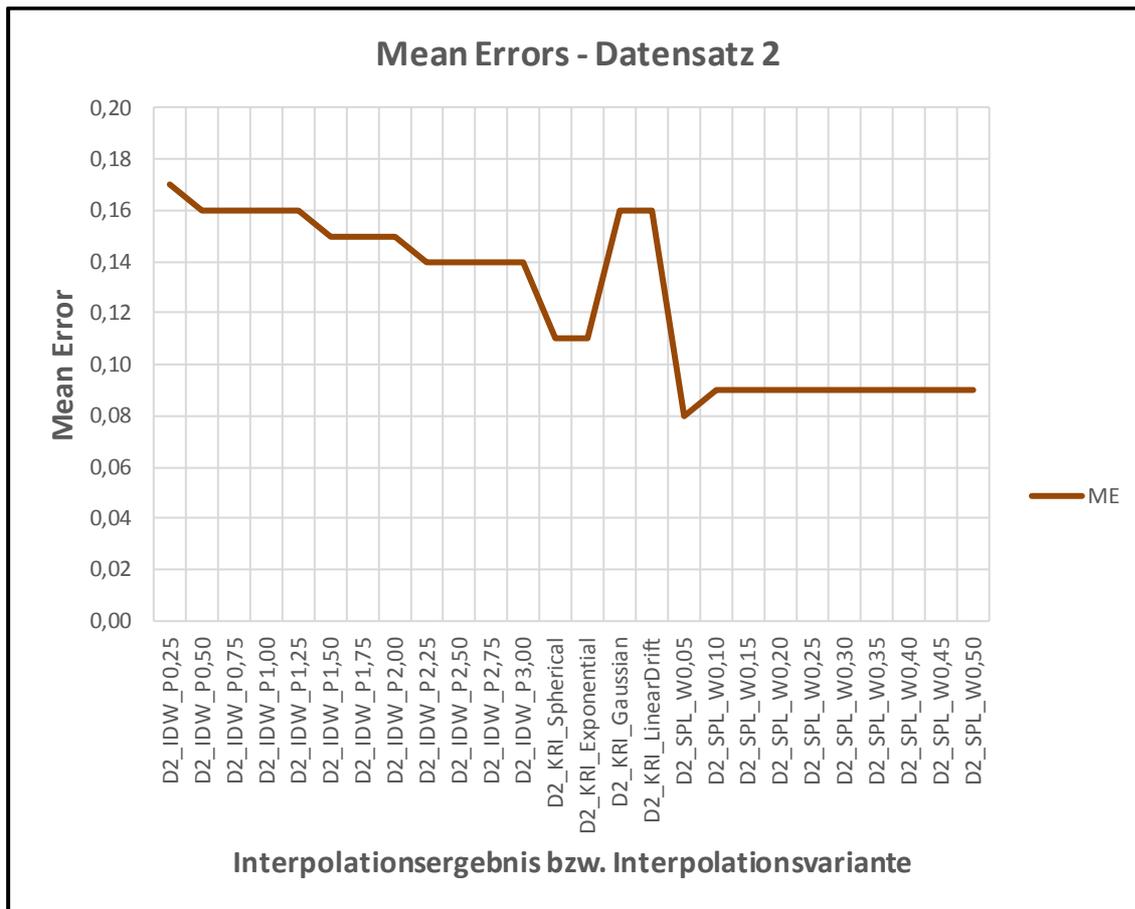


Abbildung 49: ermittelte „Mean Errors“ – Basisdatensatz 2

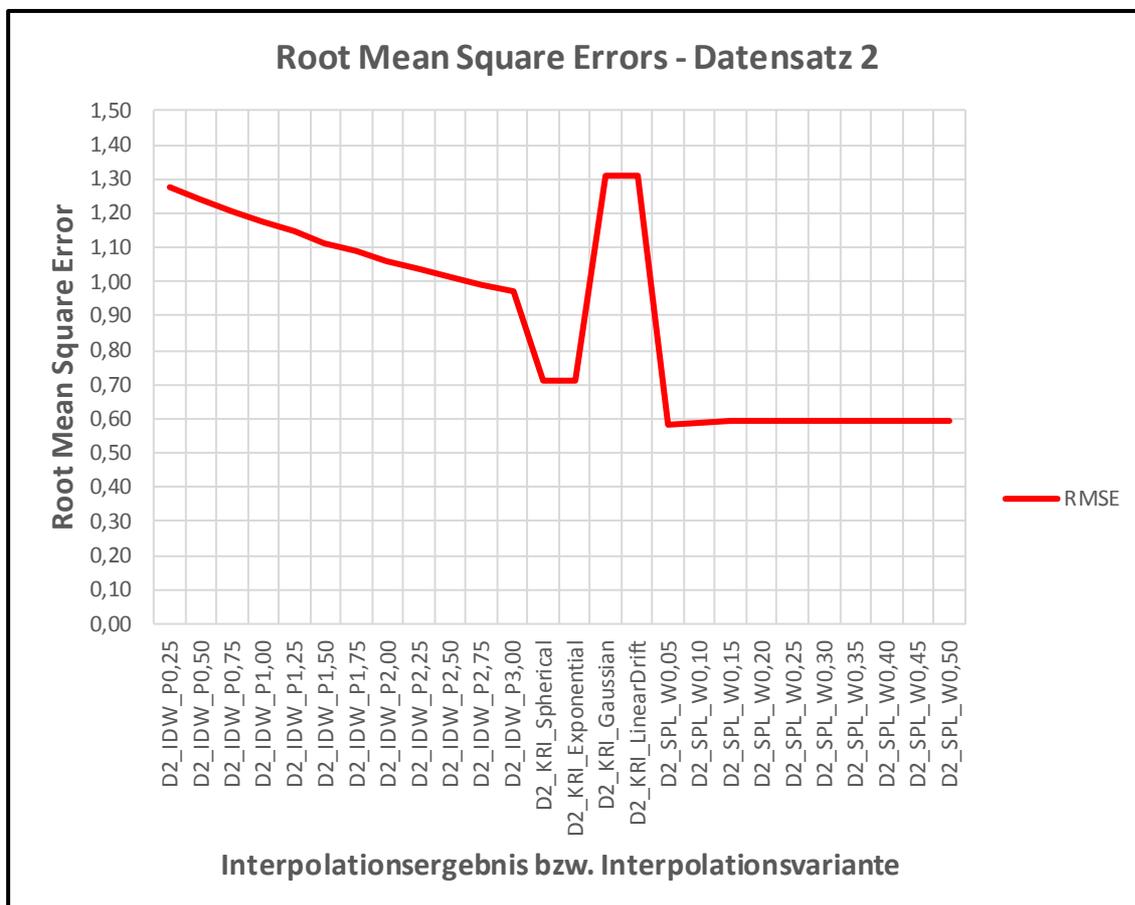


Abbildung 50: ermittelte „Root Mean Square Errors“ – Basisdatensatz 2

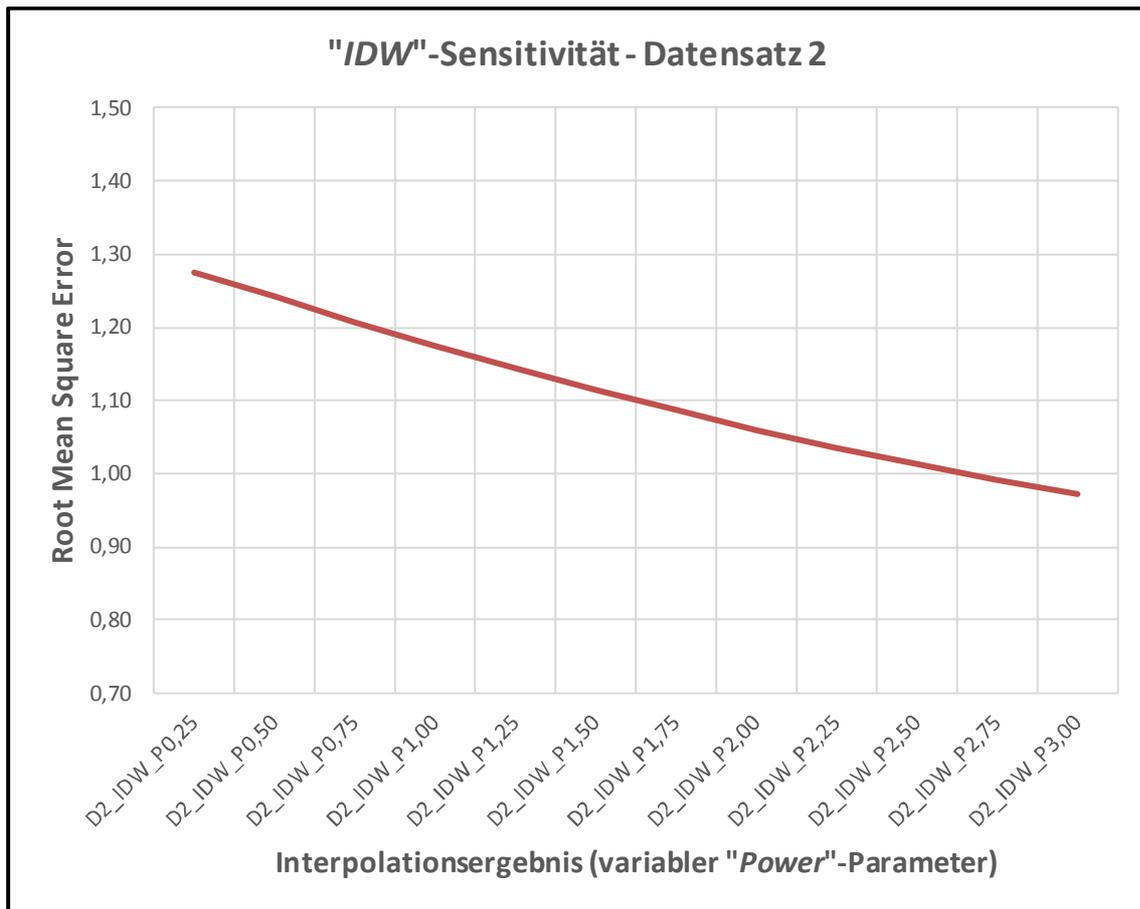


Abbildung 51: „IDW“-Sensitivität – Basisdatensatz 2

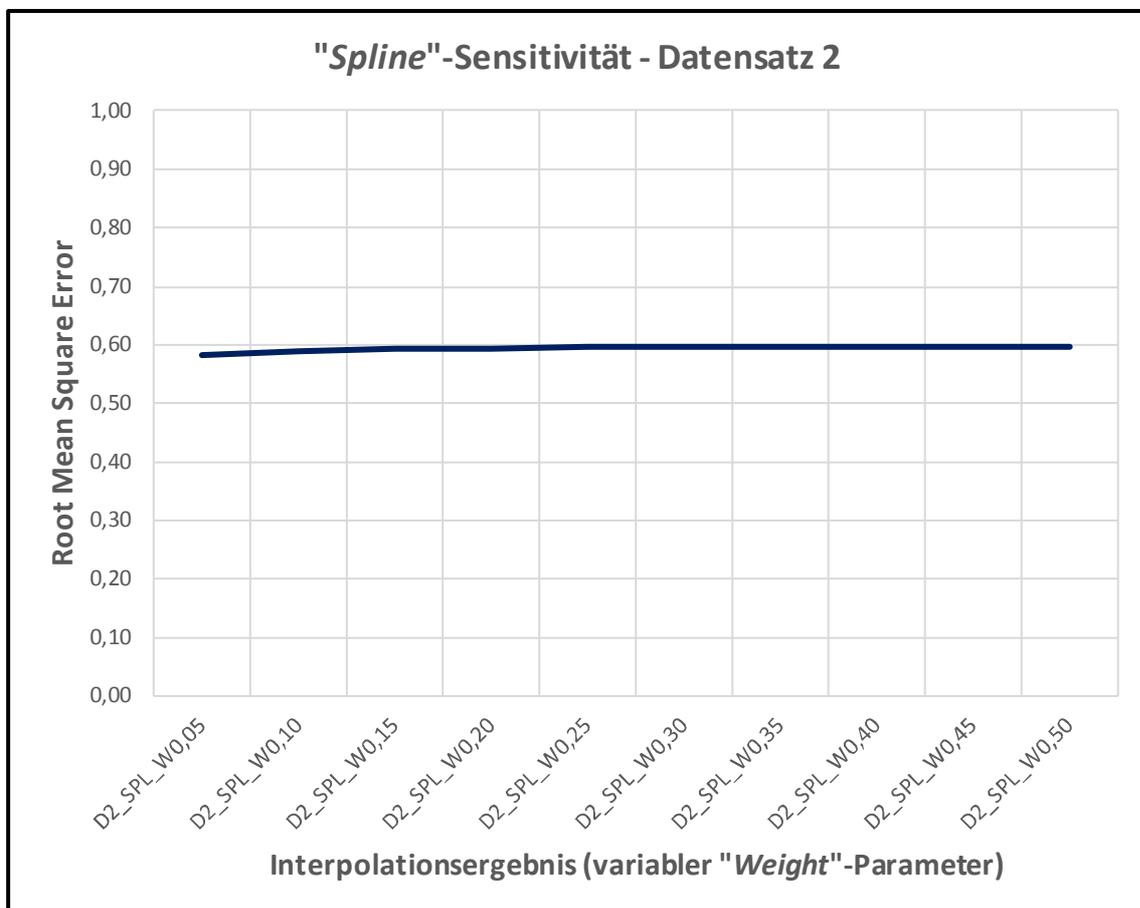


Abbildung 52: „Spline“-Sensitivität – Basisdatensatz 2

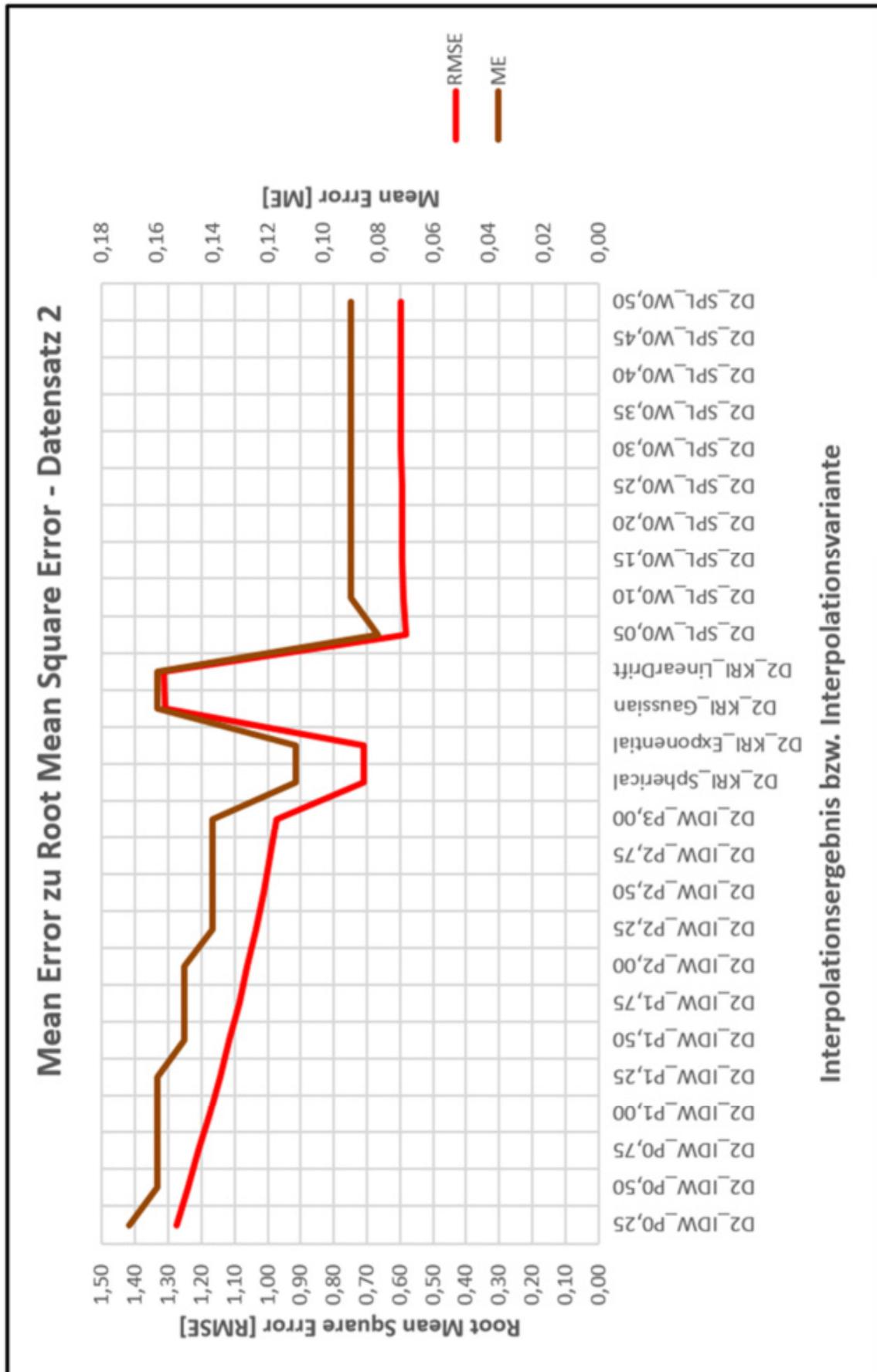


Abbildung 53: „Mean Error“ zu „Root Mean Square Error“ – Basisdatensatz 2

## 5.3.3 Ergebnisse – Basisdatensatz 3

Die nachfolgende Tabelle 13 zeigt die, bezogen auf den Basisdatensatz 3, ermittelten Fehlerwerte „Mean Errors“ und „Root Mean Square Errors“ für alle berechneten Interpolationsoberflächen, wobei auch die Sensitivität des „IDW“-Verfahrens anhand des variablen „Power“-Parameters [Parameterraum: 0,25; 0,50; 0,75; 1,00; 1,25; 1,50; 1,75; 2,00; 2,25; 2,50; 2,75; 3,0], und beim „Spline“-Verfahren anhand eines variablen „Weight“-Parameters [Parameterraum: 0,05; 0,10; 0,15; 0,20; 0,25; 0,30; 0,35; 0,40; 0,45; 0,50] dargestellt wird.

Ergebnis	Power / Weight / Var.	Datensatz	Algorithmus	ME	RMSE
D3_IDW_P0,25	0,25	3	IDW	3,83	10,76
D3_IDW_P0,50	0,50	3	IDW	3,72	10,57
D3_IDW_P0,75	0,75	3	IDW	3,60	10,38
D3_IDW_P1,00	1,00	3	IDW	3,49	10,20
D3_IDW_P1,25	1,25	3	IDW	3,38	10,02
D3_IDW_P1,50	1,50	3	IDW	3,27	9,85
D3_IDW_P1,75	1,75	3	IDW	3,16	9,69
D3_IDW_P2,00	2,00	3	IDW	3,05	9,53
D3_IDW_P2,25	2,25	3	IDW	2,95	9,39
D3_IDW_P2,50	2,50	3	IDW	2,85	9,25
D3_IDW_P2,75	2,75	3	IDW	2,76	9,13
D3_IDW_P3,00	3,00	3	IDW	2,67	9,02
D3_KRI_Spherical	Spherical	3	Kriging	3,72	10,63
D3_KRI_Exponential	Exponential	3	Kriging	1,59	6,93
D3_KRI_Gaussian	Gaussian	3	Kriging	3,95	10,95
D3_KRI_LinearDrift	LinearDrift	3	Kriging	3,81	11,06
D3_SPL_W0,05	0,05	3	Spline	0,10	3,59
D3_SPL_W0,10	0,10	3	Spline	0,06	3,54
D3_SPL_W0,15	0,15	3	Spline	0,05	3,53
D3_SPL_W0,20	0,20	3	Spline	0,04	3,52
D3_SPL_W0,25	0,25	3	Spline	0,04	3,51
D3_SPL_W0,30	0,30	3	Spline	0,03	3,51
D3_SPL_W0,35	0,35	3	Spline	0,03	3,50
D3_SPL_W0,40	0,40	3	Spline	0,03	3,50
D3_SPL_W0,45	0,45	3	Spline	0,03	3,50
D3_SPL_W0,50	0,50	3	Spline	0,03	3,50

Tabelle 13: Ergebnisse der Kreuzvalidierung („ME“ und „RMSE“) – Basisdatensatz 3

Die Abbildungen 54 und 55 stellen die, bezogen auf den Basisdatensatz 3, ermittelten Fehlerwerte anhand von Liniendiagrammen dar, wobei jede Fehlerkennzahl separat als eigenes Diagramm dargestellt wird. Weiters wird anhand der beiden Abbildungen 56 bis 57 jeweils die Sensitivität des „IDW“- bzw. „Spline“-Verfahrens gezeigt und anhand der resultierenden „RMSEs“ visualisiert.

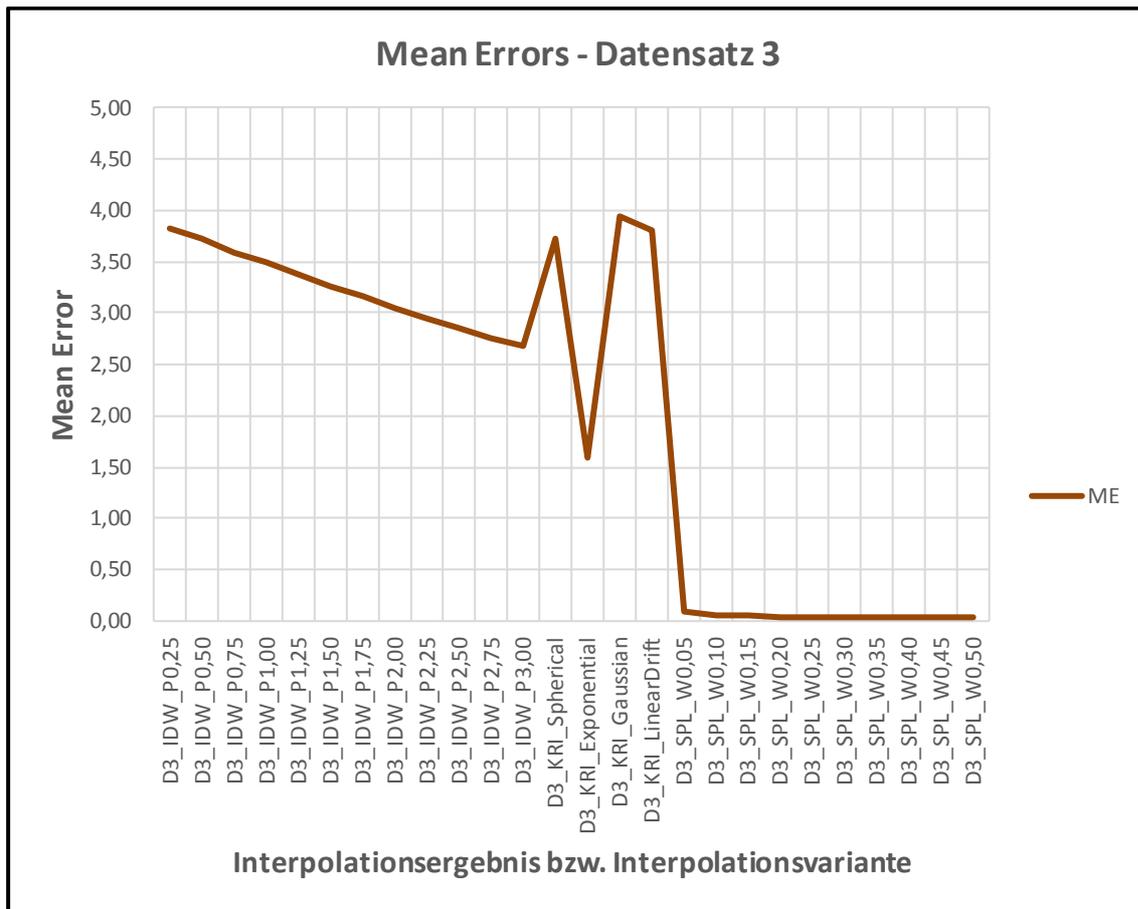


Abbildung 54: ermittelte „Mean Errors“ – Basisdatensatz 3

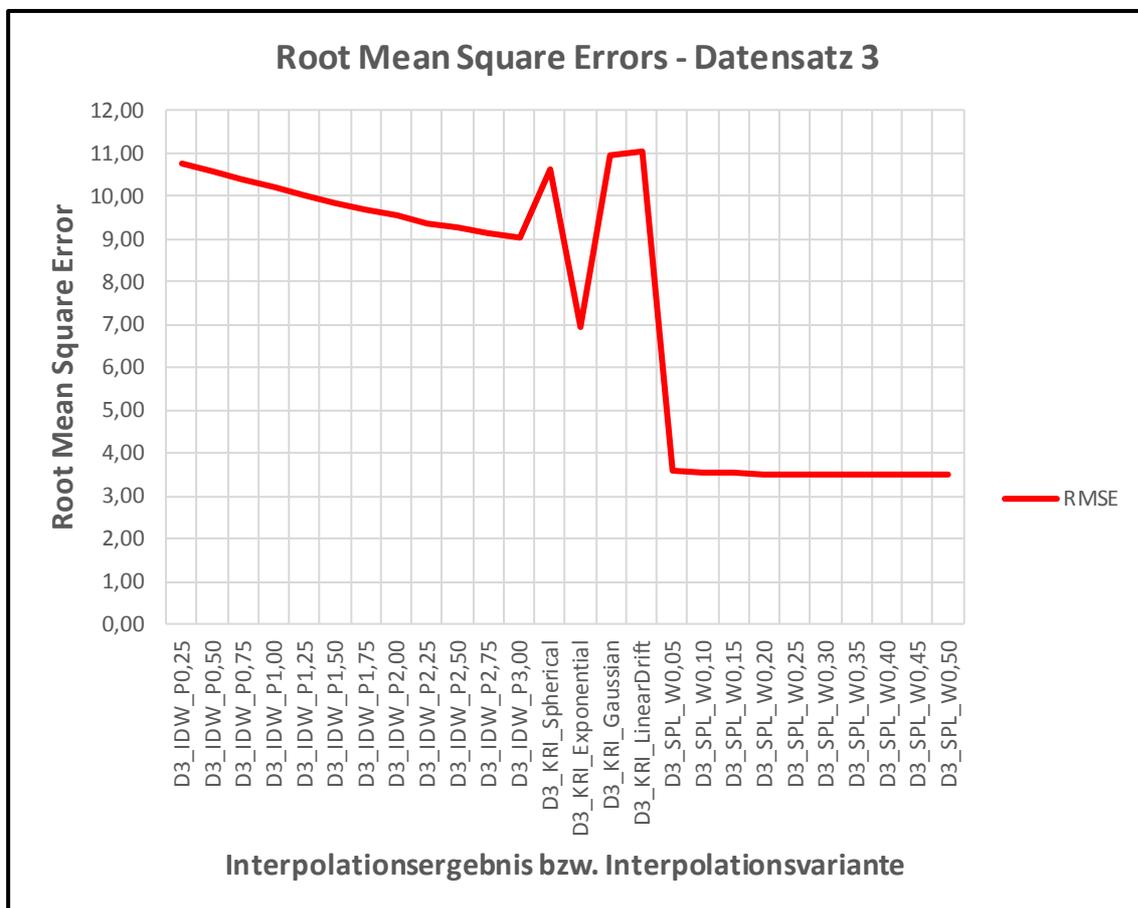


Abbildung 55: ermittelte „Root Mean Square Errors“ – Basisdatensatz 3

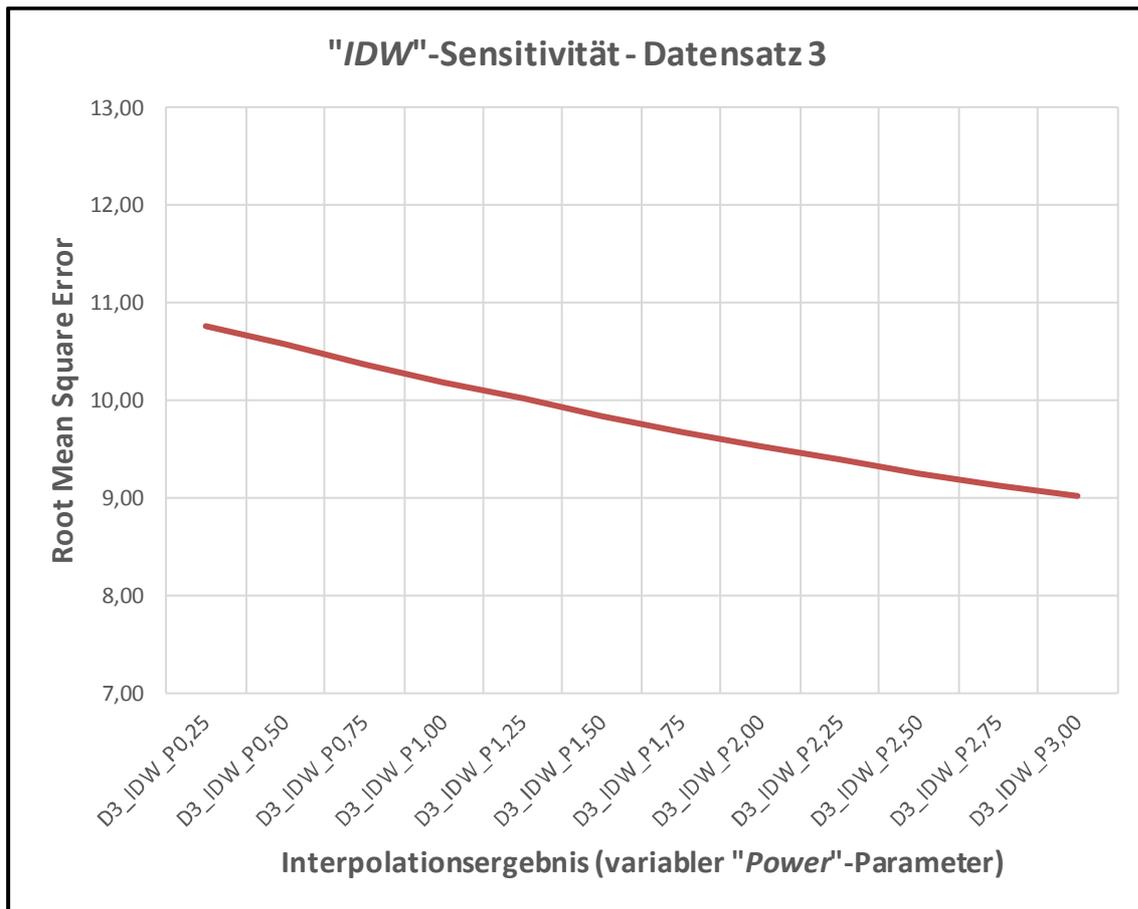


Abbildung 56: „IDW“-Sensitivität – Basisdatensatz 3

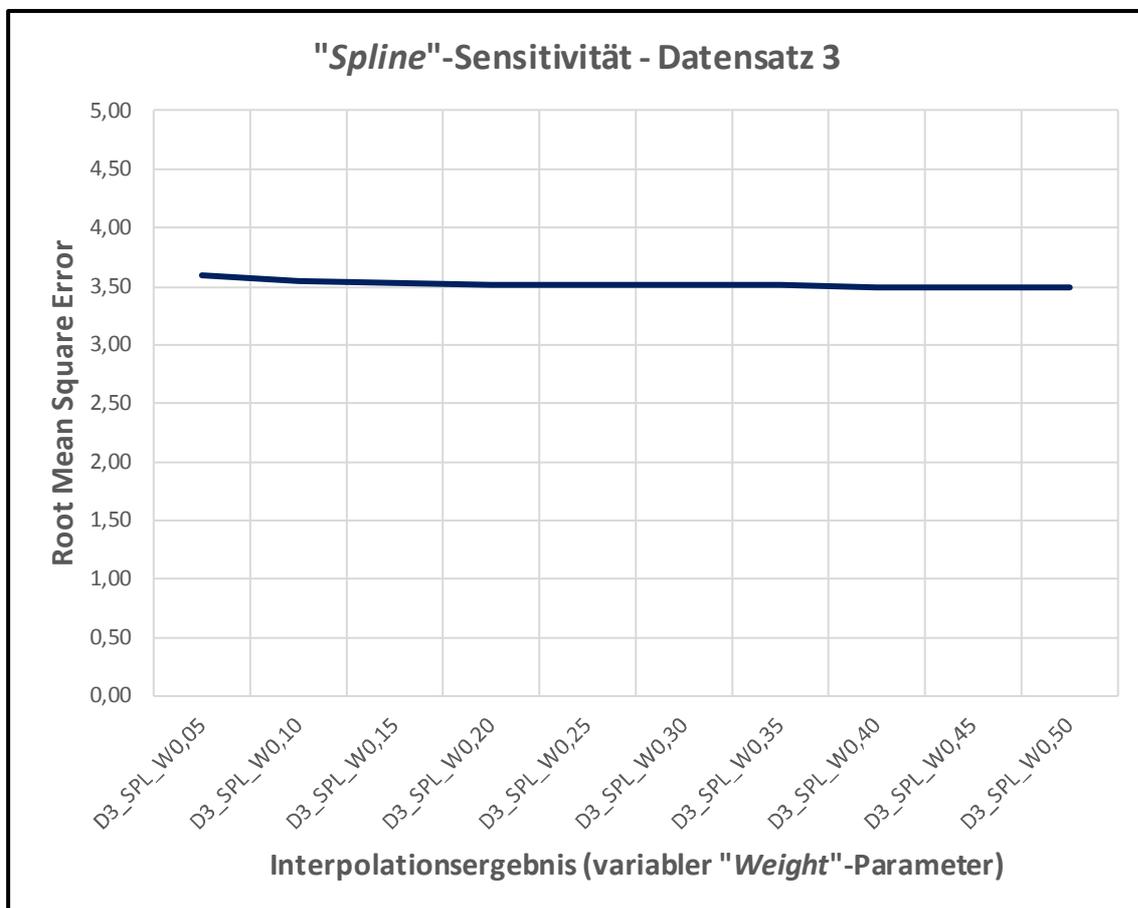


Abbildung 57: „Spline“-Sensitivität – Basisdatensatz 3

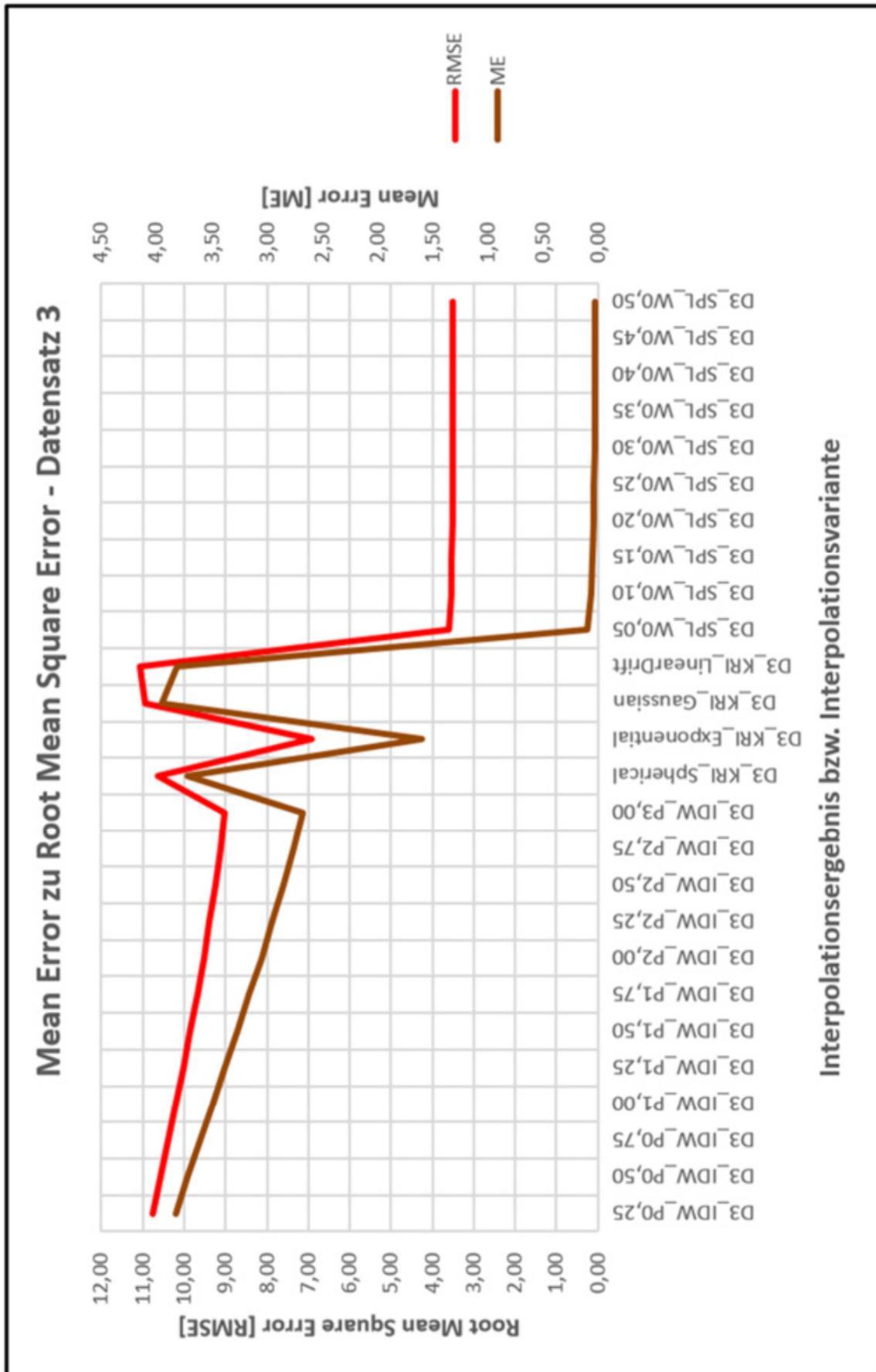


Abbildung 58: „Mean Error“ zu „Root Mean Square Error“ – Basisdatensatz 3

## 6. Diskussion

Folgende drei Forschungsfragen sollten anhand der vorliegenden Arbeit beantwortet werden:

1. **Welche Verfahren zur Interpolation von digitalen Oberflächen gibt es?**
2. **Ergeben sich bei der Gegenüberstellung der drei gängigsten Interpolationsalgorithmen Unterschiede in Bezug auf deren Interpolationsergebnisse?**
3. **Welche dieser drei Interpolationsalgorithmen liefert hinsichtlich der Topographie und der Messpunktdichte die besten Ergebnisse?**

Für die Erstellung von digitalen Oberflächenmodellen stehen grundsätzlich unterschiedliche Interpolationsverfahren zur Verfügung, wobei diese in zwei große Gruppen unterteilt werden können.

Zu den **deterministischen Interpolationsverfahren** zählen dabei:

- „Inverse Distance Weighted“ („IDW“)
- „Global Polynomial Interpolation“ („GPI“)
- „Local Polynomial Interpolation“ („LPI“)
- „Natural Neighbor“ („NN“)
- „Radial Basis Function“ („RBF“)
- „Spline“

Zu den **probabilistischen bzw. geostatistischen Interpolationsverfahren** zählt hauptsächlich das „Kriging“-Verfahren, wobei dieses in folgende (Haupt-)Varianten unterteilt werden kann:

- „Ordinary Kriging“ („OK“)
- „Universal Kriging“ („UK“)
- „Simple Kriging“ („SK“)
- „Indicator Kriging“ („IK“)
- „Co-Kriging“
- „Fuzzy Kriging“
- „Empirical Bayesian Kriging“ („EBK“)

Zu den drei gängigsten und am häufigst verwendeten Algorithmen gehören dabei die „Inverse Distance Weighted“-Interpolation, das „Kriging“-Verfahren und die „Spline“-Interpolation. In der vorliegenden Arbeit werden diese drei Interpolationsverfahren miteinander verglichen, wobei beim „Kriging“-Verfahren die Varianten „Ordinary Kriging“ mit den Semivariogramm-Modellen „Spherical“, „Exponential“ bzw. „Gaussian“ sowie „Universal Kriging“ („Linear Drift“) untersucht wurden. Bei der „Spline“-Interpolation wurde die Variante „Regularized-Spline“ verwendet.

Der wissenschaftliche Vergleich erbrachte anhand der durchgeführten Kreuzvalidierungen aller Interpolationsergebnisse, und somit auf Basis der, für alle erstellten Interpolationsoberflächen, ermittelten Fehlerkennwerte („*Mean Errors*“ bzw. „*Root Mean Square Errors*“) folgende Erkenntnisse:

- **Resultate bzw. Erkenntnisse bezüglich des Basisdatensatzes 1**

Aufgrund der sehr hohen Datendichte und der Topographie des 5 x 5 Meter Punkterasters, in Form einer gleichmäßigen Punkteverteilung, unterscheiden sich die jeweils ermittelten Fehlerwerte („*Mean Errors*“ bzw. „*Root Mean Square Errors*“) nur geringfügigst voneinander. Alle drei Interpolationsverfahren („*IDW*“, „*Kriging*“ bzw. „*Spline*“) und die in diesem Zusammenhang verwendeten Interpolationsvarianten liefern nahezu idente Resultate, wobei die berechneten Fehlerwerte grundsätzlich sehr gering ausfallen. Dies lässt sich hauptsächlich mit der sehr hohen Auflösung der verwendeten Eingangsdaten erklären.

Die Sensitivitätsanalyse beim „*IDW*“-Algorithmus (Abbildung 46) zeigt, dass die ermittelten „*Root Mean Square Errors*“ über den gesamten Parameterraum des „*Power*“-Parameters konstant bleiben und daher das „*IDW*“-Verfahren, auf Basis des verwendeten Datensatzes, keine nachweisbare Sensitivität zeigt. Auch beim „*Spline*“-Algorithmus (Abbildung 47) bleiben die ermittelten „*RMSEs*“ über den gesamten Parameterraum des „*Weight*“-Parameters konstant. Beide Verfahren reagieren somit, hinsichtlich der Eingabe unterschiedlicher Werte bezogen auf deren (Haupt-)Eingabeparameter, nicht sensitiv, wobei dies nach Auffassung des Erstellers der vorliegenden Arbeit an der Struktur des verwendeten Datensatzes liegt. Neben den jeweiligen (Haupt-)Eingabeparameter wurde bei beiden Interpolationsverfahren noch eine variierende Eingabe weiterer Interpolationsparameter getestet, wobei jedoch keine relevante Änderung der damit erzeugten Fehlerwerte beobachtet werden konnte. Anhand des verwendeten Basisdatensatzes 1 kann letztlich keine eindeutige Aussage hinsichtlich der am besten geeigneten Interpolationsmethode getroffen werden, wobei jedoch ersichtlich ist, dass bei Untersuchungsgebieten mit sehr hohen Datendichten bzw. Auflösungen grundsätzlich jedes der drei Interpolationsverfahren zur Berechnung der Interpolationsoberflächen verwendet werden kann, wobei mit keinen signifikanten Unterschieden bezüglich der daraus resultierenden Fehlerwerte zu rechnen ist.

- **Resultate bzw. Erkenntnisse bezüglich des Basisdatensatzes 2**

Bei diesem Datensatz unterscheiden sich die je nach Fehlerkennzahl („*Mean Error*“ bzw. „*Root Mean Square Error*“) berechneten Fehlerwerte deutlich voneinander. Die mit Hilfe der Kreuzvalidierung ermittelten Werte zeigen, dass hier der „*Spline*“-Algorithmus jeweils die niedrigsten (Gesamt-)Fehler generiert. Aufgrund der geringeren Datendichte des 25 x 25 Meter Punkterasters, in Form einer gleichmäßigen Punkteverteilung, fallen die ermittelten Fehlerwerte bereits deutlich höher aus als dies beim Basisdatensatz 1 der Fall ist.

Die Sensitivitätsanalyse beim „IDW“-Algorithmus (Abbildung 51) zeigt, dass dieses Verfahren, auf Basis des verwendeten Datensatzes, durchaus sensitiv auf die Eingabe unterschiedlicher Werte bezogen auf den „Power“-Parameter reagiert. Je höher dieser (Haupt-)Eingabeparameter gewählt wurde desto geringer fielen die Fehlerwerte („ME“ bzw. „RMSE“) aus. Der vordefinierte Parameterraum des „Power“-Parameters und die daraus resultierenden Fehlerwerte lassen den Schluss zu, dass dieses Interpolationsverfahren grundsätzlich sensitiv reagiert, und dies bei dessen Verwendung berücksichtigt werden sollte. Der „Spline“-Algorithmus (Abbildung 52) hingegen reagiert nur unwesentlich auf die Eingabe unterschiedlicher Werte bezüglich des „Weight“-Parameters und zeigt somit keine relevante Empfindlichkeit hinsichtlich des verwendeten Parameterraumes. Die graphische Gegenüberstellung (Abbildung 53) der berechneten „Mean Errors“ zu den „Root Mean Square Errors“ in Form eines Liniendiagramms zeigt, dass die beiden Fehlerkennzahlen einen sehr ähnlichen Kurvenverlauf bzw. Trend aufweisen.

Anhand des Basisdatensatzes 2 konnte festgestellt werden, dass hier der „Spline“-Interpolationsalgorithmus, hinsichtlich der Datendichte und der Topographie, die geringsten Fehlerwerte und somit die besten Interpolationsergebnisse liefert. Auch das „Ordinary Kriging“-Verfahren mit den Semivariogramm-Modellen „Spherical“ bzw. „Exponential“ schneidet im Vergleich gut ab, wobei diese Interpolationsmethoden jedoch jeweils höhere Fehlerwerte als beim „Spline“-Verfahren zur Folge haben.

- **Resultate bzw. Erkenntnisse bezüglich des Basisdatensatzes 3**

Die Resultate des dritten Basisdatensatzes bestätigen die, hinsichtlich des zweiten Datensatzes, gewonnenen Erkenntnisse. Wiederum zeigen die mit Hilfe der durchgeführten Kreuzvalidierung ermittelten Fehlerwerte, dass der „Spline“-Algorithmus jeweils die deutlich niedrigsten Fehlerwerte aller berechneten Interpolationsoberflächen generiert. Da die Datendichte des 78 x 78 Meter Punkterasters, ebenfalls in Form einer gleichmäßigen Punkteverteilung, am geringsten ist und topographisch betrachtet zum Teil große Höhenunterschiede aufweist, fallen die ermittelten Fehlerwerte diesbezüglich wie erwartet wiederum höher aus als diese bei den Basisdatensätzen 1 und 2.

Die Sensitivitätsanalyse beim „IDW“-Algorithmus (Abbildung 56) und beim „Spline“-Algorithmus (Abbildung 57) zeigen die gleichen grundlegenden Kurvenverläufe wie die vom vorherigen Basisdatensatz, wobei wie bereits erwähnt die Fehlerwerte jeweils höher ausfallen. Auch die graphische Gegenüberstellung (Abbildung 58) der berechneten „MEs“ zu den „RMSEs“ bestätigt die bereits gewonnenen Erkenntnisse.

Anhand des dritten und letzten Basisdatensatzes konnte somit ebenfalls festgestellt werden, dass der „Spline“-Interpolationsalgorithmus, auf Basis der ermittelten Fehlerwerte und des verwendeten Testdatensatzes, letztlich die besten Interpolationsergebnisse liefert.

Im Vergleich zu den Ergebnissen der im Literaturüberblick dargestellten wissenschaftlichen Studien bzw. Publikationen zeigen die Resultate der vorliegenden Arbeit, dass nicht wie gewissermaßen erwartet das „*Kriging*“-Verfahren, als komplexes geostatistisches Interpolationsverfahren, im Vergleich am besten abschneidet, sondern letztendlich der „*Spline*“-Interpolationsalgorithmus.

In den Publikationen von ZIMMERMAN et al. (1999), AZOUZI & SCHRÖTER (2012), ARUN (2013) und ADHIKARY & DASH (2017) generierten, hinsichtlich der ermittelten Fehlerkennzahlen, jeweils eine bzw. mehrere unterschiedliche „*Kriging*“-Interpolationsvarianten am Ende die geringsten Fehlerwerte. Dabei wurden neben dem „*Inverse Distance Weighted*“- bzw. dem „*Spline*“-Algorithmus zum Teil auch noch weitere deterministische Verfahren wie z.B. „*ANUDEM*“, die „*Natural Neighbor*“-Methode oder die „*Radial Basis Function*“ mit den jeweils verwendeten „*Kriging*“-Varianten verglichen. Für die einzelnen Vergleiche wurden dabei unterschiedlichste Testdatensätze herangezogen, wobei am häufigsten „*Shuttle Radar Topography Mission*“-Daten der jeweiligen Untersuchungsgebiete zur Anwendung kamen.

Bei der wissenschaftlichen Untersuchung von PAVLOVA (2017) hingegen lieferten die beiden „*Spline*“-Varianten „*Regularized-Spline*“ bzw. „*Tension-Spline*“ die besten Interpolationsergebnisse, nachdem insgesamt sechs verschiedene Interpolationsalgorithmen verglichen wurden.

QIAO et al. (2018) verglichen das „*Inverse Distance Weighted*“- und das „*Ordinary Kriging*“-Verfahren und zeigten, dass bei diesem speziellen Anwendungsfall der „*IDW*“-Interpolationsalgorithmus, aufgrund der sehr geringen Anzahl von lediglich 146 Eingangsdatenpunkten, am besten für die räumliche Vorhersage der Schwermetallbelastung in den Böden von Peking geeignet ist.

Somit kann, auf Basis all dieser bereits durchgeführten Vergleiche von Interpolationsalgorithmen bzw. Verfahren keine eindeutige Aussage hinsichtlich der am besten geeigneten Interpolationsvariante gemacht werden, wobei das „*Kriging*“-Verfahren grundsätzlich öfter als die anderen Interpolationsverfahren die besten Resultate lieferte. Bei der Gegenüberstellung der jeweils ermittelten Fehlerwerte zeigt sich jedoch, dass sich die Resultate vom „*Kriging*“- und vom „*Spline*“-Algorithmus zum Teil nur geringfügig voneinander unterscheiden. Auch in der vorliegenden Arbeit lagen die Ergebnisse des „*Kriging*“-Verfahrens mit den Semivariogramm-Modellen „*Spherical*“ bzw. „*Exponential*“ zum Teil nahe bei denen der „*Spline*“-Interpolation. Die schlechtesten Ergebnisse lieferte letztlich der „*IDW*“-Interpolationsalgorithmus, wobei hier die berechneten Fehlerwerte im Durchschnitt immer größer ausfielen wie bei den anderen beiden Interpolationsverfahren.

Grundsätzlich konnten alle in der vorliegenden Arbeit zu bearbeitenden Forschungsfragen anhand der angewandten Methodik, und somit zur Erstellung, Gegenüberstellung und Beurteilung der unterschiedlichen Interpolationsoberflächen, welche anhand der drei gängigsten Interpolationsalgorithmen „*IDW*“, „*Kriging*“ und „*Spline*“ erstellt wurden, vollständig beantwortet werden.

## 7. Schlussfolgerungen und Ausblick

Das Ziel der Master Thesis ist der Vergleich der drei am häufigst verwendeten Interpolationsalgorithmen bzw. Verfahren „*Inverse Distance Weighted*“, „*Kriging*“ und „*Spline*“, welche grundsätzlich zur Erstellung von digitalen Oberflächenmodellen verwendet werden. Hierfür wurden, auf Basis drei unterschiedlicher Testdatensätze, für jede der Interpolationsmethoden jeweils mehrere Interpolationsoberflächen erstellt und diese anschließend miteinander verglichen. Beim „*IDW*“- bzw. „*Spline*“-Algorithmus wurde des Weiteren anhand von vordefinierten Parameterräumen die jeweilige Sensitivität des Interpolationsverfahrens getestet, wobei lediglich bei der „*IDW*“-Methode eine solche letztlich nachgewiesen werden konnte. Das Verfahren reagierte dabei deutlich sensitiv auf die Eingabe unterschiedlicher „*Power*“-Werte und somit auf dessen Haupteingabeparameter. Je höher dieser Gewichtungparameter der Distanz gewählt wurde desto geringer fielen die ermittelten Fehlerwerte aus. Das „*Spline*“-Verfahren hingegen reagierte nur unwesentlich auf die Eingabe unterschiedlicher Werte bezüglich des „*Weight*“-Parameters.

Für die Analyse bzw. Beurteilung der Ergebnisse wurde neben einer visuellen Darstellung und einer Explorativen Räumlichen Datenanalyse („*ESDA*“) auch eine Kreuzvalidierung auf Basis der Ermittlung des „*Mean Errors*“ und des „*Root Mean Square Errors*“ durchgeführt, wobei diese beiden Fehlerkennzahlen für alle Interpolationsoberflächen berechnet wurden.

Die nachfolgende Tabelle 14 fasst alle Interpolationsergebnisse hinsichtlich der jeweils ermittelten Fehlerwerte zusammen, wobei diese wiederum nach Basisdatensatz 1 bis 3 getrennt visualisiert werden. Es wird dabei der minimale und zudem maximale „*Mean Error*“ („*ME*“) bzw. „*Root Mean Square Error*“ („*RMSE*“) je Interpolationsvariante bzw. Verfahren dargestellt:

Interpolationsalgorithmus	ME-Min.	ME-Max.	RMSE-Min.	RMSE-Max.
<b>Interpolationsergebnisse - Basisdatensatz 1</b>				
IDW	0,05	0,06	0,45	0,45
Kriging	0,04	0,06	0,44	0,45
Spline	0,02	0,02	0,44	0,44
<b>Interpolationsergebnisse - Basisdatensatz 2</b>				
IDW	0,14	0,17	0,97	1,28
Kriging	0,11	0,16	0,71	1,31
Spline	0,08	0,09	0,58	0,60
<b>Interpolationsergebnisse - Basisdatensatz 3</b>				
IDW	2,67	3,83	9,02	10,76
Kriging	1,59	3,95	6,93	11,06
Spline	0,03	0,10	3,50	3,59

Tabelle 14: Gegenüberstellung der ermittelten Fehlerwerte („*ME*“ bzw. „*RMSE*“)

Zur Bestätigung und Validierung der aus den verwendeten Basisdatensätzen 1 bis 3 gewonnenen Erkenntnisse wurden am Ende noch zwei zusätzliche Testdatensätze untersucht, wobei hierfür wieder jeweils gleichmäßige Punkteverteilungen mit einer Auflösung von jeweils 50 x 50 Meter verwendet wurden. Die beiden Untersuchungsgebiete lagen dabei einerseits im nördlichen Weinviertel in Niederösterreich und andererseits in der Umgebung von Parndorf im Bezirk Neusiedl am See im Burgenland. Somit wurde wiederum eine andere Auflösung bzw. Datendichte der Punkt(mess)daten als Eingangsdaten, sowie zwei weitere unterschiedliche topographische Situationen als Grundlage für die Erstellung der Interpolationsoberflächen und der anschließenden Berechnung der diesbezüglichen Fehlerwerte verwendet. Anhand der daraus ermittelten Fehlerwerte zeigte sich dabei ein weiteres Mal, dass der „Spline“-Interpolationsalgorithmus die besten Ergebnisse lieferte und somit konnten die bereits gewonnenen Erkenntnisse bestätigt werden.

Zusammengefasst konnte anhand dieser Master Thesis und der, für den Vergleich der drei Interpolationsalgorithmen bzw. Verfahren „*Inverse Distance Weighted*“, „*Kriging*“ und „*Spline*“, verwendeten Basisdatensätze festgestellt werden, dass in Hinblick auf die Topographie und Datendichte letztlich in allen betrachteten Fällen die „Spline“-Interpolationsmethode die jeweils geringsten Fehlerwerte („*Mean Error*“ bzw. „*Root Mean Square Error*“) bei den durchgeführten Kreuzvalidierungen lieferte. Auch die visuelle und explorative räumliche Datenanalyse der erstellten Interpolationsoberflächen bestätigt dabei die gewonnenen Erkenntnisse.

Inwieweit die, auf Basis der verwendeten Punkteverteilungen, getätigten Aussagen und Schlussfolgerungen auch für andere abweichende räumliche Punktemuster (z.B. zufällige bzw. konzentrierte Verteilungen) zutreffen, kann aus Sicht des Verfassers der vorliegenden Arbeit erst in zukünftigen weiterführenden wissenschaftlichen Studien eindeutig geklärt werden.

Grundsätzlich werden Interpolationsalgorithmen auch in Zukunft eine wichtige Rolle im Zusammenhang mit dem Erzeugen von digitalen Gelände- bzw. Oberflächenmodellen, sowie der Erstellung und Darstellung unterschiedlichster Arten von Konzentrationsoberflächen wie u.a. von Niederschlagsoberflächen, der Schwermetallbelastung in Böden bzw. im Grundwasser sowie der räumlichen Vorhersage der Ozonbelastung spielen.

Die Auswahl des, für den jeweiligen Anwendungsfalles, am besten geeigneten Interpolationsverfahrens sollte dabei sorgfältig überlegt getroffen werden, wobei anhand dieser Master Thesis gezeigt wird, dass das „Spline“-Verfahren im Vergleich zu den beiden anderen Interpolationsverfahren „*IDW*“ bzw. „*Kriging*“ letztlich bessere Resultate liefert und somit durchaus als erste Wahl verwendet werden kann.

## Literaturverzeichnis

- ADHIKARY, P.P. & DASH, C.J. (2017):** Comparison of deterministic and stochastic methods to predict spatial variation of groundwater depth. – *Applied Water Science* 7, 1, 339–348, <https://link.springer.com/article/10.1007/s13201-014-0249-8> (Stand: 2014-11-16) (Zugriff: 2020-02-01).
- ARUN, P.V. (2013):** A comparative analysis of different DEM interpolation methods. – *The Egyptian Journal of Remote Sensing and Space Science* 16, 2, 133–139, <https://www.sciencedirect.com/science/article/pii/S1110982313000276> (Stand: 2013-10-08) (Zugriff: 2020-02-01).
- AZOUZI, B. & SCHRÖTER, J. (2012):** Kartenerstellung mit Geostatistik und Fuzzy-Geostatistik: Ein Methodenvergleich am Beispiel der Verteilung der Sulfatkonzentrationen im oberflächennahen Grundwasserleiter in Bremen Nord. – *Grundwasser – Zeitschrift der Fachsektion Hydrogeologie* 17, 4, 231–241, <https://link.springer.com/article/10.1007/s00767-012-0210-4> (Stand: 2012-10-23) (Zugriff: 2020-02-01).
- BARTELME, N. (2005<sup>4</sup>):** Geoinformatik: Modelle Strukturen Funktionen. Berlin, Heidelberg: Springer.
- BECKER, T., HERRMANN, R., SANDOR, V., SCHÄFER, D. & WELLISCH, U. (2016):** Stochastische Risikomodellierung und statistische Methoden: Ein anwendungsorientiertes Lehrbuch für Aktuarer. Berlin, Heidelberg: Springer.
- BOURIER, G. (2018<sup>13</sup>):** Beschreibende Statistik: Praxisorientierte Einführung - Mit Aufgaben und Lösungen. Wiesbaden: Springer Fachmedien Wiesbaden GmbH.
- BURROUGH, P.A. & MCDONNELL, R.A. (1998):** Principles of Geographical Information Systems: Spatial Information Systems and Geostatistics. New York: Oxford University Press Inc.
- DAYA, A.A. & BEJARI, H. (2015):** A comparative study between simple kriging and ordinary kriging for estimating and modeling the Cu concentration in Chehlkureh deposit, SE Iran. – *Arabian Journal of Geosciences* 8, 8, 6003–6020, <https://link.springer.com/article/10.1007/s12517-014-1618-1> (Stand: 2014-10-10) (Zugriff: 2020-02-01).
- EISANK, C. & KOCH, A. (2017a):** Modul Geostatistik - Deterministische Interpolation (2017). UNIGIS MSc Lehrgang Modul 8 Lektion 12. Interfakultärer Fachbereich für Geoinformatik - Z\_GIS der Universität Salzburg.
- EISANK, C. & KOCH, A. (2017b):** Modul Geostatistik - Probabilistische Interpolation (2017). UNIGIS MSc Lehrgang Modul 8 Lektion 13. Interfakultärer Fachbereich für Geoinformatik - Z\_GIS der Universität Salzburg.
- EISANK, C. & KOCH, A. (2017c):** Modul Geostatistik - Validierung (2017). UNIGIS MSc Lehrgang Modul 8 Lektion 14. Interfakultärer Fachbereich für Geoinformatik - Z\_GIS der Universität Salzburg.

- ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2016):** How Kriging works | ArcGIS for Desktop, <http://desktop.arcgis.com/de/arcmap/10.3/tools/3d-analyst-toolbox/how-kriging-works.htm> (Stand: 2016-04-21) (Zugriff: 2020-02-01).
- ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2018a):** How Contouring works | ArcGIS Pro, <https://pro.arcgis.com/en/pro-app/tool-reference/3d-analyst/how-contouring-works.htm> (Stand: 2018-06-10) (Zugriff: 2020-02-01).
- ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2018b):** How IDW works | ArcGIS Pro, <https://pro.arcgis.com/en/pro-app/tool-reference/3d-analyst/how-idw-works.htm> (Stand: 2018-06-10) (Zugriff: 2020-02-01).
- ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2018c):** How Intersect works | ArcGIS Pro, <https://pro.arcgis.com/en/pro-app/tool-reference/analysis/how-intersect-analysis-works.htm> (Stand: 2018-06-10) (Zugriff: 2020-02-01).
- ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2018d):** How Project Raster works | ArcGIS Pro, <https://pro.arcgis.com/en/pro-app/tool-reference/data-management/how-project-raster-works.htm> (Stand: 2018-06-10) (Zugriff: 2020-02-01).
- ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2018e):** How Spline works | ArcGIS Pro, <https://pro.arcgis.com/de/pro-app/tool-reference/3d-analyst/how-spline-works.htm> (Stand: 2018-06-10) (Zugriff: 2020-02-01).
- ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (2019):** Cross Validation | ArcGIS Pro, <https://pro.arcgis.com/de/pro-app/tool-reference/geostatistical-analyst/cross-validation.htm> (Stand: 2019-08-01) (Zugriff: 2020-02-01).
- FISHER, P.F. (1993):** Algorithm and implementation uncertainty in viewshed analysis. – International journal of geographical information systems 7, 4, 331–347, <https://www.tandfonline.com/doi/abs/10.1080/02693799308901965> (Stand: 2007-02-01) (Zugriff: 2020-02-01).
- KRIGE, D.G. (1966):** Two-dimensional weighted moving average trend surfaces for ore valuation. – Mathematical statistics and computer applications in ore valuation, a symposium, Journ. South African Inst. Mining and Metallurgy (1966), 13–79 (Zugriff: 2020-02-01).
- LANG, S. & BLASCHKE, T. (2007<sup>1</sup>):** Landschaftsanalyse mit GIS. UTB Geowissenschaften, Biologie, Ökologie, Agrar- und Forstwissenschaften UTB 8347. Stuttgart: Ulmer (UTB).
- LU, G.Y. & WONG, D.W. (2008):** An adaptive inverse-distance weighting spatial interpolation technique. – Computers & Geosciences 34, 9, 1044–1055, <https://www.sciencedirect.com/science/article/pii/S0098300408000721> (Stand: 2008-03-18) (Zugriff: 2020-02-01).
- NIELSEN, P. (2010):** WindPRO 2.7 Handbuch, [http://www.emd.dk/files/windpro/manuals/for\\_print/DE\\_WindPRO2.7\\_complete.pdf](http://www.emd.dk/files/windpro/manuals/for_print/DE_WindPRO2.7_complete.pdf) (Stand: 2019-01-07) (Zugriff: 2020-02-01).

- OTTMANN, T. & WIDMAYER, P. (2017<sup>6</sup>):** Algorithmen und Datenstrukturen. Berlin, Germany: Springer Verlag GmbH Deutschland.
- PAVLOVA, A.I. (2017):** Analysis of elevation interpolation methods for creating digital elevation models. – *Optoelectronics, Instrumentation and Data Processing* 53, 2, 171–177, <https://link.springer.com/article/10.3103/S8756699017020108> (Stand: 2017-06-16) (Zugriff: 2020-02-01).
- QIAO, P., LEI, M., YANG, S., YANG, J., GUO, G. & ZHOU, X. (2018):** Comparing ordinary kriging and inverse distance weighting for soil as pollution in Beijing. – *Environmental Science and Pollution Research* 25, 16, 15597–15608, <https://link.springer.com/article/10.1007/s11356-018-1552-y> (Stand: 2018-03-23) (Zugriff: 2020-02-01).
- SHEPARD, D. (1968):** A two-dimensional interpolation function for irregularly-spaced data. In: BLUE, R.B. & ROSENBERG, A.M. (Hrsg.). *Proceedings of the 1968 23rd ACM national conference on* -. New York, USA: ACM Press, 517–524.
- TATEOSIAN, L. (2015<sup>1</sup>):** Python For ArcGIS. Cham, s.l.: Springer International Publishing.
- ULFERTS, L. (2017):** Python mit ArcGIS: Einstieg in die Automatisierung der Geoverarbeitung in ArcGIS. Berlin, Offenbach: Wichmann.
- WRIEDT, G., VRIES, D. de, EDEN, T. & FEDEROLF, C. (2019):** Regionalisierte Darstellung der Nitratbelastung im Grundwasser Niedersachsens. – *Grundwasser – Zeitschrift der Fachsektion Hydrogeologie* 24, 1, 27–41, <https://link.springer.com/article/10.1007/s00767-019-00415-0> (Stand: 2019-01-31) (Zugriff: 2020-02-01).
- XIAO, Y., GU, X., YIN, S., SHAO, J., CUI, Y., ZHANG, Q. & NIU, Y. (2016):** Geostatistical interpolation model selection based on ArcGIS and spatio-temporal variability analysis of groundwater level in piedmont plains, northwest China. – *SpringerPlus* 5, 425, <https://link.springer.com/article/10.1186/s40064-016-2073-0> (Stand: 2016-04-11) (Zugriff: 2020-02-01).
- ZIMMERMAN, D., PAVLIK, C., RUGGLES, A. & ARMSTRONG, M.P. (1999):** An Experimental Comparison of Ordinary and Universal Kriging and Inverse Distance Weighting. – *Mathematical Geology* 31, 4, 375–390, <https://link.springer.com/article/10.1023/A:1007586507433> (Zugriff: 2020-02-01).

## Anhang A – ArcGIS Pro Python Add-In

```

# Import System Modules
import arcpy
import os
import time
import arcpy.analysis
import arcpy.management

# Clear Screen
os.system("CLS")

# Overwrite Function
arcpy.env.overwriteOutput = True

# Workspace Definition
arcpy.env.workspace = "C:\EW-10535-MT\OUTPUT-ArcGIS-Pro"

print("")
print("")
print("")
print("-----")
print("----- Name:      ArcGIS-Pro_Python_Add-In.py -----")
print("----- Version:    1.0.3 -----")
print("----- Created by:  Erwin Winkler (UNIGIS-ID:10535) -----")
print("----- Created on:  28.10.2019 - 12:00 -----")
print("----- Created for: ESRI ArcGIS Pro 2.X.X -----")
print("-----")
print("")
print("")
print("-----")
print("Dieses Python Add-In fuer ArcGIS Pro erzeugt aus einem Hoehenraster als Eingangsdatenquelle mit Hilfe")
print("einer Sequenz von 4 (Geo-)Algorithmen einen frei definierten Ausschnitt von Hoehenlinien (Isohypsen)")
print("als Ausgangsdaten. Die Eingangsdatenquelle und Werkzeugparameter sind dabei grossteils frei wahlbar.")
print("Der Vorgabewert kann jeweils mit ->ENTER<- bestätigt werden oder die Eingabe erfolgt manuell. Weiters")
print("ist es moeglich beim 1.Algorithmus den Interpolationsalgorithmus (IDW, Kriging, Spline) auszuwaehlen.")
print("")
print("")
print("Der Programmablauf ist folgendermaßen definiert:")
print("-----")
print("")
print("Schritt 1: Eingangsdaten (Source-Data) -----> auswaehlen")
print("Schritt 2: Algorithmus/Tool 1 - IDW/Kriging/Spline ---> Interpolationsalgorithmus/Parameter festlegen")
print("Schritt 3: Algorithmus/Tool 2 - Project-Raster -----> Parameter festlegen")
print("Schritt 4: Algorithmus/Tool 3 - Contour -----> Parameter festlegen")
print("Schritt 5: Algorithmus/Tool 4 - Intersect -----> Parameter festlegen")
print("Schritt 6: Berechnungen/Interpolationsoberflaechen -----> werden durchgefuehrt/erzeugt")
print("")
print("")
print("Schematische Darstellung des Berechnungsablaufes:")
print("-----")
print("")
print("I-----I I-----I I-----I I-----I I-----I I-----I")
print("I I IDW I I I I I I I I I")
print("I Source I---->I Kriging I---->I Project I---->I Contour I---->I Intersect I---->I Target I")
print("I -Data I I Spline I I -Raster I I I I I -Data I")
print("I-----I I-----I I-----I I-----I I-----I I-----I")
print("")
print("")
print("-----")
print("")
print("")
print("")
pressEnterToNext = input("-----> Um Fortzufahren bitte die ENTER-Taste druecken <-----")

# Clear Screen
os.system("CLS")

# Source-Data Definition
print("")
print("")
print("")
print("-----")
print("-----> Eingangsdaten (Source-Data) <-----")
print("-----")
print("")
print("")
print("-----")
print("1... fuer --> Basisdatensatz 1: 5 x 5 Meter BEV-Raster --> C:\EW-10535-MT\INPUT-DATA\BEV-5x5m.shp")
print("2... fuer --> Basisdatensatz 2: 25x25 Meter BEV-Raster --> C:\EW-10535-MT\INPUT-DATA\BEV-25x25m.shp")
print("3... fuer --> Basisdatensatz 3: 90x90 Meter SRTM-Raster --> C:\EW-10535-MT\INPUT-DATA\SRTM-90x90m.shp")
print("-----")
print("")
Source_Question = input("Wollen Sie den Eingangsdatenpfad manuell eingeben [ja/nein]? ")
print("")
print("")
if Source_Question == 'ja':
    IN_File = input("Bitte den Eingangsdatenpfad (Source-Data) jetzt eingeben: ")
else:
    Datensatz_Auswahl = input("Bitte waehlen Sie den gewünschten Basisdatensatz [1 / 2 / 3] aus: ")

    if Datensatz_Auswahl == "1":
        IN_File = "C:\EW-10535-MT\INPUT-DATA\BEV-5x5m.shp"
        DATENSATZ = "5x5m"

    if Datensatz_Auswahl == "2":
        IN_File = "C:\EW-10535-MT\INPUT-DATA\BEV-25x25m.shp"
        DATENSATZ = "25x25m"

```

```

if Datensatz_Auswahl == "3":
    IN_File = "C:\EW-10535-MT\INPUT-DATA\SRM-90x90m.shp"
    DATENSATZ = "SRM-90x90m"

if Datensatz_Auswahl == "":
    IN_File = "C:\EW-10535-MT\INPUT-DATA\SRM-90x90m.shp"
    DATENSATZ = "SRM-90x90m"

print("")
print("")
print("Folgende Eingangsdaten werden fuer die Berechnungen verwendet:")
print("-----")
print(IN_File)
print("")
print("")
print("-----")
print("")
print("")
print("")
print("-----> Bitte einen Moment warten.... <-----")

# Time delay
time.sleep(3)
# Clear Screen
os.system("CLS")

# Algorithm Declaration for Algorithm/Tool 1
print("")
print("")
print("")
print("-----")
print("-----> Auswahl der Interpolationsmethode fuer Algorithmus/Tool 1 <-----")
print("-----")
print("")
print("")
print("-----")
print("1... fuer ---> IDW <---")
print("2... fuer ---> Kriging <---")
print("3... fuer ---> Spline <---")
print("-----")
InterpolationAuswahl = input("Bitte waehlen Sie die gewuenschte Interpolationsmethode [1 / 2 / 3] ")
print("")
print("")
print("Folgende Interpolationsmethode wird fuer alle nun folgenden Berechnungen verwendet:")
print("-----")
if InterpolationAuswahl == "1":
    print("-----> IDW <-----")
elif InterpolationAuswahl == "2":
    print("-----> Kriging <-----")
elif InterpolationAuswahl == "3":
    print("-----> Spline <-----")
else:
    InterpolationAuswahl = "1"
    print("-----> Es wird das Standard-Interpolationsverfahren ---> IDW <--- verwendet! <-----")
print("")
print("")
print("-----")
print("-----")
print("")
print("")
print("-----> Bitte einen Moment warten.... <-----")

# Time delay
time.sleep(7)
# Clear Screen
os.system("CLS")

if InterpolationAuswahl == "1":
    # Variable Declaration for Algorithm/Tool 1 - "IDW"
    print("")
    print("")
    print("")
    print("-----")
    print("-----> IDW-Parameter <-----")
    print("-----")
    print("")
    print("")
    Z_Value_Field = input("Z Value Field (->Elevation<-) mit ENTER bestaetigen bzw. jetzt eingeben: ")
    print("")
    if Z_Value_Field == "":
        Z_Value_Field = "Elevation"
    while True:
        try:
            Cell_Size_1 = input("Output Cell Size (->1<-) mit ENTER bestaetigen bzw. jetzt eingeben: ")
            if Cell_Size_1 == "":
                Cell_Size_1 = "1"
            Cell_Size_1 = int(Cell_Size_1)
            break
        except ValueError:
            print("")
            print("ACHTUNG ----- ACHTUNG")
            print("ACHTUNG -----> Sie haben keine Zahl (Integer) eingegeben ---> Bitte Eingabe wiederholen! <---- ACHTUNG")
            print("ACHTUNG ----- ACHTUNG")
            print("")
    print("")
    Search_Radius = input("Search Radius (->VARIABLE 12<-) mit ENTER bestaetigen bzw. jetzt eingeben: ")
    if Search_Radius == "":
        Search_Radius = "VARIABLE 12"
    print("")

```

```

print("")
print("")
print("Folgende IDW-Interpolationsparameter werden fuer die Berechnung verwendet:")
print("-----")
print("")
print("Z Value Field: ")
print(Z_Value_Field)
print("")
print("Output Cell Size: ")
print(Cell_Size_1)
print("")
print("Power: ")
print("0,25 / 0,5 / 0,75 / 1 / 1,25 / 1,5 / 1,75 / 2 / 2,25 / 2,5 / 2,75 / 3")
print("")
print("Search Radius: ")
print(Search_Radius)
print("")
print("")
print("-----")
print("-----")
print("")
print("")
print("")
print("-----> Bitte einen Moment warten.... <-----")
# Time delay
time.sleep(7)
# Clear Screen
os.system("CLS")

if InterpolationAuswahl == "2":
    # Variable Declaration for Algorithm/Tool 1 - "Kriging"
    print("")
    print("")
    print("")
    print("-----")
    print("-----> Kriging-Parameter <-----")
    print("-----")
    print("")
    Z_Field_Kriging = input("Z Value Field (->Elevation<-) mit ENTER bestaetigen bzw. jetzt eingeben: ")
    print("")
    if Z_Field_Kriging == "":
        Z_Field_Kriging = "Elevation"
    while True:
        try:
            Cell_Size_Kriging = input("Output Cell Size (->1<-) mit ENTER bestaetigen bzw. jetzt eingeben: ")
            if Cell_Size_Kriging == "":
                Cell_Size_Kriging = "1"
            Cell_Size_Kriging = int(Cell_Size_Kriging)
            break
        except ValueError:
            print("")
            print("ACHTUNG ----- ACHTUNG")
            print("ACHTUNG ----> Sie haben keine Zahl (Integer) eingegeben ----> Bitte Eingabe wiederholen! <---- ACHTUNG")
            print("ACHTUNG ----- ACHTUNG")
            print("")
    print("")
    Search_Radius_Kriging = input("Search Radius (->VARIABLE 12<-) mit ENTER bestaetigen bzw. jetzt eingeben: ")
    if Search_Radius_Kriging == "":
        Search_Radius_Kriging = "VARIABLE 12"
    print("")
    print("")
    print("")
    print("Die Berechnungen werden fuer folgende Kriging-Methoden durchgefuehrt:")
    print("-----")
    print("")
    print("----> Ordinary    -> Spherical <-----")
    print("----> Ordinary    -> Exponential <----")
    print("----> Ordinary    -> Gaussian <-----")
    print("----> Universal    -> Linear Drift <----")
    print("")
    print("")
    print("")
    print("Folgende Kriging-Interpolationsparameter werden dabei fuer alle Berechnungen verwendet:")
    print("-----")
    print("")
    print("Z Value Field: ")
    print(Z_Field_Kriging)
    print("")
    print("Output Cell Size: ")
    print(Cell_Size_Kriging)
    print("")
    print("Search Radius: ")
    print(Search_Radius_Kriging)
    print("")
    print("")
    print("-----")
    print("-----")
    print("")
    print("")
    print("")
    print("-----> Bitte einen Moment warten.... <-----")
    # Time delay
    time.sleep(7)
    # Clear Screen
    os.system("CLS")

if InterpolationAuswahl == "3":
    # Variable Declaration for Algorithm/Tool 1 - "Spline"
    print("")
    print("")
    print("")

```

```

print("-----")
print("-----> Spline-Parameter <-----")
print("-----")
print("")
print("")
SPLINE_Z_Value_Field = input("Z Value Field (->Elevation<-) mit ENTER bestaetigen bzw. jetzt eingeben: ")
print("")
if SPLINE_Z_Value_Field == "":
    SPLINE_Z_Value_Field = "Elevation"
while True:
    try:
        SPLINE_Output_Cell_Size = input("Output Cell Size (->1<-) mit ENTER bestaetigen bzw. jetzt eingeben: ")
        if SPLINE_Output_Cell_Size == "":
            SPLINE_Output_Cell_Size = "1"
        SPLINE_Output_Cell_Size = int(SPLINE_Output_Cell_Size)
        break
    except ValueError:
        print("")
        print("ACHTUNG ----- ACHTUNG")
        print("ACHTUNG ----> Sie haben keine Zahl (Integer) eingegeben ----> Bitte Eingabe wiederholen! <---- ACHTUNG")
        print("ACHTUNG ----- ACHTUNG")
        print("")
print("")
SPLINE_Type = input("Spline Type (->REGULARIZED<-) mit ENTER bestaetigen bzw. jetzt eingeben: ")
if SPLINE_Type == "":
    SPLINE_Type = "REGULARIZED"
print("")
while True:
    try:
        SPLINE_Number_of_Points = input("Number of Points (->12<-) mit ENTER bestaetigen bzw. jetzt eingeben: ")
        if SPLINE_Number_of_Points == "":
            SPLINE_Number_of_Points = "12"
        SPLINE_Number_of_Points = int(SPLINE_Number_of_Points)
        break
    except ValueError:
        print("")
        print("ACHTUNG ----- ACHTUNG")
        print("ACHTUNG ----> Sie haben keine Zahl (Integer) eingegeben ----> Bitte Eingabe wiederholen! <---- ACHTUNG")
        print("ACHTUNG ----- ACHTUNG")
        print("")
print("")
print("")
print("Folgende Spline-Interpolationsparameter werden fuer die Berechnung verwendet:")
print("-----")
print("")
print("Z Value Field: ")
print(SPLINE_Z_Value_Field)
print("")
print("Output Cell Size: ")
print(SPLINE_Output_Cell_Size)
print("")
print("Weight: ")
print("0,05 / 0,1 / 0,15 / 0,2 / 0,25 / 0,3 / 0,35 / 0,4 / 0,45 / 0,5")
print("")
print("Spline Type: ")
print(SPLINE_Type)
print("")
print("Number of Points: ")
print(SPLINE_Number_of_Points)
print("")
print("")
print("-----")
print("-----")
print("")
print("")
print("")
print("-----> Bitte einen Moment warten.... <-----")
# Time delay
time.sleep(7)
# Clear Screen
os.system("CLS")

# Variable Declaration for Algorithm/Tool 2 - "Project Raster"
print("")
print("")
print("")
print("-----")
print("-----> Project Raster-Parameter <-----")
print("-----")
print("")
print("")
Geographic_Transformation = input("Geogr. Transfor. (->MGI_To_ETRS_1989_1<-) mit ENTER bestaetigen bzw. eingeben: ")
print("")
if Geographic_Transformation == "":
    Geographic_Transformation = "MGI_To_ETRS_1989_1"
while True:
    try:
        Cell_Size_2 = input("Output Cell Size (->1<-) mit ENTER bestaetigen bzw. jetzt eingeben: ")
        if Cell_Size_2 == "":
            Cell_Size_2 = "1"
        Cell_Size_2 = int(Cell_Size_2)
        break
    except ValueError:
        print("")
        print("ACHTUNG ----- ACHTUNG")
        print("ACHTUNG ----> Sie haben keine Zahl (Integer) eingegeben ----> Bitte Eingabe wiederholen! <---- ACHTUNG")
        print("ACHTUNG ----- ACHTUNG")
        print("")
print("")
Resampling_Technique = input("Resampling Technique (->NEAREST<-) mit ENTER bestaetigen bzw. jetzt eingeben: ")
if Resampling_Technique == "":

```

```

Resampling_Technique = "NEAREST"
Coordinate_System_Input = "PROJCS['MGI_Transverse_Mercator',GEOGCS['GCS_MGI',DATUM['D_MGI',SPHEROID['Bes-
sel_1841',6377397.155,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Transverse_Mercator'],PARAME-
TER['False_Easting',0.0],PARAMETER['False_Northing',-5000000.0],PARAMETER['Central_Meridian',13.3333333],PARAMETER['Scale_Fac-
tor',1.0],PARAMETER['Latitude_Of_Origin',0.0],UNIT['Meter',1.0]]"
Coordinate_System_Output = "PROJCS['ETRS_1989_UTM_Zone_33N',GEOGCS['GCS_ETRS_1989',DATUM['D_ETRS_1989',SPHE-
ROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Transverse_Merca-
tor'],PARAMETER['False_Easting',500000.0],PARAMETER['False_Northing',0.0],PARAMETER['Central_Meridian',15.0],PARAMETER['Scale_Fac-
tor',0.9996],PARAMETER['Latitude_Of_Origin',0.0],UNIT['Meter',1.0]]"
print("")
print("")
print("")
print("Folgende Project Raster-Parameter werden fuer die Berechnung verwendet:")
print("-----")
print("")
print("Input Coordinate System: ")
print("MGI Transverse Mercator GK M31 - EPSG:31255")
print("")
print("Output Coordinate System: ")
print("ETRS 1989 UTM Zone 33N - EPSG:25833")
print("")
print("Geographic Transformation: ")
print(Geographic_Transformation)
print("")
print("Output Cell Size: ")
print(Cell_Size_2)
print("")
print("Resampling Technique: ")
print(Resampling_Technique)
print("")
print("-----")
print("-----")
print("")
print("")
print("")
print("-----> Bitte einen Moment warten... <-----")

# Time delay
time.sleep(3)
# Clear Screen
os.system("CLS")

# Variable Declaration for Algorithm/Tool 3 - "Contour"
print("")
print("")
print("")
print("-----")
print("-----> Contour-Parameter <-----")
print("-----")
print("")
print("")
while True:
    try:
        Contour_Interval = input("Contour Interval (->1<-) mit ENTER bestaetigen bzw. jetzt eingeben: ")
        if Contour_Interval == "":
            Contour_Interval = "1"
        Contour_Interval = int(Contour_Interval)
        break
    except ValueError:
        print("")
        print("ACHTUNG ----- ACHTUNG")
        print("ACHTUNG ----> Sie haben keine Zahl (Integer) eingegeben ----> Bitte Eingabe wiederholen! <---- ACHTUNG")
        print("ACHTUNG ----- ACHTUNG")
        print("")
print("")
while True:
    try:
        Base_Contour = input("Base Contour (->0<-) mit ENTER bestaetigen bzw. jetzt eingeben: ")
        if Base_Contour == "":
            Base_Contour = "0"
        Base_Contour = int(Base_Contour)
        break
    except ValueError:
        print("")
        print("ACHTUNG ----- ACHTUNG")
        print("ACHTUNG ----> Sie haben keine Zahl (Integer) eingegeben ----> Bitte Eingabe wiederholen! <---- ACHTUNG")
        print("ACHTUNG ----- ACHTUNG")
        print("")
print("")
while True:
    try:
        Z_Factor = input("Z Factor (->1<-) mit ENTER bestaetigen bzw. jetzt eingeben: ")
        if Z_Factor == "":
            Z_Factor = "1"
        Z_Factor = int(Z_Factor)
        break
    except ValueError:
        print("")
        print("ACHTUNG ----- ACHTUNG")
        print("ACHTUNG ----> Sie haben keine Zahl (Integer) eingegeben ----> Bitte Eingabe wiederholen! <---- ACHTUNG")
        print("ACHTUNG ----- ACHTUNG")
        print("")
print("")
print("")
print("")
print("Folgende Contour-Parameter werden fuer die Berechnung verwendet:")
print("-----")
print("")
print("Contour Interval: ")
print(Contour_Interval)
print("")

```

```

print("Base Contour: ")
print(Base_Contour)
print("")
print("Z Factor: ")
print(Z_Factor)
print("")
print("-----")
print("-----")
print("")
print("")
print("-----> Bitte einen Moment warten... <-----")

# Time delay
time.sleep(3)
# Clear Screen
os.system("CLS")

# Variable Declaration for Algorithm/Tool 4 - "Intersect"
print("")
print("")
print("")
print("-----")
print("-----> Auswahlfenster fuer Intersect-Operation <-----")
print("-----")
print("")
print("")
print("-----")
print("Vorgabewert: C:\EW-10535-MT\LIMITATION-AREA\limitation-Area.shp")
print("-----")
print("mit Projektion: ETRS1989 UTM Zone 33N -> EPSG:25833")
print("-----")
print("")
Area_Question = input("Wollen Sie das Auswahlfenster in Form eines Polygon-Shapefiles (xxx.shp) manuell eingeben [ja/nein]? ")
print("")
if Area_Question == 'ja':
    Selection_Area = input("Bitte den Dateipfad fuer das Polygon-Shapefile jetzt eingeben: ")
else:
    Selection_Area = "C:\EW-10535-MT\LIMITATION-AREA\Limitation-Area.shp"
print("")
print("")
print("Folgendes Auswahlfenster wird verwendet:")
print("-----")
print(Selection_Area)
print("")
print("")
print("-----")
print("-----")
print("")
print("")
print("-----> Bitte einen Moment warten... <-----")

# Process with IDW as Algorithm/Tool 1
# -----
if InterpolationAuswahl == "1":
    # Interim Results Definition
    ZE1oE = "C:\EW-10535-MT\OUTPUT-ArcGIS-Pro\ZE1_IDW-Power_"
    ZE1E = ".tif"
    ZE2PROJECToE = "C:\EW-10535-MT\OUTPUT-ArcGIS-Pro\ZE2_PROJECT_mit_IDW-Power_"
    ZE2PROJECTE = ".tif"
    ZE3CONTOURoE = "C:\EW-10535-MT\OUTPUT-ArcGIS-Pro\ZE3_CONTOUR_mit_IDW-Power_"
    ZE3CONTOURE = ".shp"
    OUT_FileoE = "C:\EW-10535-MT\OUTPUT-ArcGIS-Pro\ERGEBNIS_mit_IDW-Power_"
    OUT_FileE = ".shp"

    IDW_Power_Bereich = ["0,25", "0,5", "0,75", "1,0", "1,25", "1,5", "1,75", "2,0", "2,25", "2,5", "2,75", "3,0"]
    for IDW_Power in IDW_Power_Bereich:

        # Time delay
        time.sleep(3)
        # Clear Screen
        os.system("CLS")
        print("")
        print("")
        print("")
        print("-----")
        print("-----> Bitte warten - Berechnungen werden durchgeführt... <-----")
        print("-----")
        print("-----> Bitte warten - Berechnungen werden durchgeführt... <-----")
        print("-----")
        print("-----> Bitte warten - Berechnungen werden durchgeführt... <-----")
        print("-----")
        print("-----> Bitte warten - Berechnungen werden durchgeführt... <-----")
        print("-----")
        print("")
        print("")
        print("")

    # Performing Algorithm/Tool 1 - "IDW"
    OUT_IDW_Tool = arcpy.sa.Idw(IN_File, Z_Value_Field, Cell_Size_1, IDW_Power, Search_Radius, None)
    OUT_IDW_Tool.save(ZE1oE + IDW_Power + ZE1E)
    print("-----> Der IDW-Algorithmus wurde mit Power-Parameter = " + IDW_Power + " erfolgreich ausgeführt...")
    print("Power-Parameter: " + IDW_Power)
    print("-----> " + ZE1oE + IDW_Power + ZE1E + " erfolgreich erstellt!")
    print("")
    print("")

```

```

# -----
---
# Raster-Properties (Statistics) for Result of Algorithm/Tool 1
# Process: Get Raster Properties --> MINIMUM
EW_MIN_Result = arcpy.GetRasterProperties_management((ZE1oE + IDW_Power + ZE1E), property_type="MINIMUM", band_index="Band_1")
EW_MIN = EW_MIN_Result.getOutput(0)
print("MIN-Wert: " + EW_MIN)
print("")
# Process: Get Raster Properties --> MAXIMUM
EW_MAX_Result = arcpy.GetRasterProperties_management((ZE1oE + IDW_Power + ZE1E), property_type="MAXIMUM", band_index="Band_1")
EW_MAX = EW_MAX_Result.getOutput(0)
print("MAX-Wert: " + EW_MAX)
print("")
# Process: Get Raster Properties --> MEAN
EW_MEAN_Result = arcpy.GetRasterProperties_management((ZE1oE + IDW_Power + ZE1E), property_type="MEAN", band_index="Band_1")
EW_MEAN = EW_MEAN_Result.getOutput(0)
print("MEAN-Wert: " + EW_MEAN)
print("")
# Process: Get Raster Properties --> STD-Deviation
EW_STD_Result = arcpy.GetRasterProperties_management((ZE1oE + IDW_Power + ZE1E), property_type="STD", band_index="Band_1")
EW_STD = EW_STD_Result.getOutput(0)
print("STD-Wert: " + EW_STD)
print("")
# Process: Get Raster Properties --> noDATA
EW_noDATA_Result = arcpy.GetRasterProperties_management((ZE1oE + IDW_Power + ZE1E), property_type="ANYNODATA", band_in-
dex="Band_1")
EW_noDATA = EW_noDATA_Result.getOutput(0)
print("Nulls-Werte: " + EW_noDATA)
print("")
# -----
---
# -----
---
# Process: Extract Result-Values To Table
In_Feature_EW = "C:\EW-10535-MT\REFERENCE-POINTS\Referencepoints.shp"
In_Raster_EW = ZE1oE + IDW_Power + ZE1E
Out_Table_EW = ZE1oE + IDW_Power + "_Table.dbf"
Out_Raster_Names_Table_EW = ""
arcpy.ExtractValuesToTable_ga(In_Feature_EW, In_Raster_EW, Out_Table_EW, Out_Raster_Names_Table_EW, "ADD_WARNING_FIELD")
print("-----> " + Out_Table_EW + " erfolgreich erstellt!")
print("")
# -----
---
# Process: Add Fields to Table
Attribute_Table_1 = float(IDW_Power.replace(",",".")) * 100
Attribute_Table_2 = int(Attribute_Table_1)
Attribute_Table_3 = str(Attribute_Table_2)
EW_field_name_1 = "IDW_P" + Attribute_Table_3
EW_field_name_2 = "D_IDW_P" + Attribute_Table_3
EW_field_name_3 = "Q_IDW_P" + Attribute_Table_3
arcpy.AddField_management(Out_Table_EW, EW_field_name_1, "DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
arcpy.AddField_management(Out_Table_EW, EW_field_name_2, "DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
arcpy.AddField_management(Out_Table_EW, EW_field_name_3, "DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
# Process: Calculate Field in Table
arcpy.CalculateField_management(Out_Table_EW, EW_field_name_1, "!Value!", "PYTHON3", "")
# Process: Delete Fields in Table
arcpy.DeleteField_management(Out_Table_EW, "Value")
arcpy.DeleteField_management(Out_Table_EW, "SrcID_Feat")
arcpy.DeleteField_management(Out_Table_EW, "SrcID_Rast")
# Process: Join Fields in Table and Reference-Table
EWS_Reference_Table = "C:\EW-10535-MT\REFERENCE-TABLES\Referencepoints_Table.dbf"
EWS_Fields = EW_field_name_1 + ";" + EW_field_name_2 + ";" + EW_field_name_3
arcpy.JoinField_management(EWS_Reference_Table, "OID", Out_Table_EW, "OID", EWS_Fields)
# Process: Calculate Field in Reference-Table
EW_Expression_1 = "!" + EW_field_name_1 + "! - !POINT_Z!"
arcpy.CalculateField_management(EWS_Reference_Table, EW_field_name_2, EW_Expression_1, "PYTHON3", "")
# Process: Calculate Field in Reference-Table
EW_Expression_2 = "!" + EW_field_name_2 + "! * !" + EW_field_name_2 + "!"
arcpy.CalculateField_management(EWS_Reference_Table, EW_field_name_3, EW_Expression_2, "PYTHON3", "")
# -----
---
# Performing Algorithm/Tool 2 - "Project Raster"
arcpy.management.ProjectRaster((ZE1oE + IDW_Power + ZE1E), (ZE2PROJECToE + IDW_Power + ZE2PROJECTE), Coordinate_System_Output,
Resampling_Technique, Cell_Size_2, Geographic_Transformation, None, Coordinate_System_Input)
print("-----> Der Project Raster-Algorithmus wurde erfolgreich ausgefuehrt...")
print("-----> " + ZE2PROJECToE + IDW_Power + ZE2PROJECTE + " erfolgreich erstellt!")
print("")
# -----
---
# Raster-Properties (Statistics) for Result of Algorithm/Tool 2
# Process: Get Raster Properties --> MINIMUM
EW_MIN_Result = arcpy.GetRasterProperties_management((ZE2PROJECToE + IDW_Power + ZE2PROJECTE), property_type="MINIMUM", band_in-
dex="Band_1")
EW_MIN = EW_MIN_Result.getOutput(0)
print("MIN-Wert: " + EW_MIN)
print("")
# Process: Get Raster Properties --> MAXIMUM
EW_MAX_Result = arcpy.GetRasterProperties_management((ZE2PROJECToE + IDW_Power + ZE2PROJECTE), property_type="MAXIMUM", band_in-
dex="Band_1")
EW_MAX = EW_MAX_Result.getOutput(0)
print("MAX-Wert: " + EW_MAX)
print("")
# Process: Get Raster Properties --> MEAN
EW_MEAN_Result = arcpy.GetRasterProperties_management((ZE2PROJECToE + IDW_Power + ZE2PROJECTE), property_type="MEAN", band_in-
dex="Band_1")
EW_MEAN = EW_MEAN_Result.getOutput(0)
print("MEAN-Wert: " + EW_MEAN)

```

```

print ("")
# Process: Get Raster Properties ---> STD-Deviation
EW_STD_Result = arcpy.GetRasterProperties_management((ZE2PROJECToE + IDW_Power + ZE2PROJECTE), property_type="STD", band_in-
dex="Band_1")
EW_STD = EW_STD_Result.getOutput(0)
print("STD-Wert: " + EW_STD)
print("")
# Process: Get Raster Properties ---> noDATA
EW_noDATA_Result = arcpy.GetRasterProperties_management((ZE2PROJECToE + IDW_Power + ZE2PROJECTE), property_type="ANYNODATA",
band_index="Band_1")
EW_noDATA = EW_noDATA_Result.getOutput(0)
print("Nulls-Werte: " + EW_noDATA)
print("")
# -----
---

# Performing Algorithm/Tool 3 - "Contour"
arcpy.sa.Contour((ZE2PROJECToE + IDW_Power + ZE2PROJECTE), (ZE3CONTOURoE + IDW_Power + ZE3CONTOURE), Contour_Interval, Base_Con-
tour, Z_Factor)
print("-----> Der Contour-Algorithmus wurde erfolgreich ausgefuehrt...")
print("-----> " + ZE3CONTOURoE + IDW_Power + ZE3CONTOURE + " erfolgreich erstellt!")
print("")
print("")

# Performing Algorithm/Tool 4 - "Intersect"
IN_INTERSECT_Tool = [(ZE3CONTOURoE + IDW_Power + ZE3CONTOURE), Selection_Area]
arcpy.analysis.Intersect(IN_INTERSECT_Tool, (OUT_FileoE + IDW_Power + OUT_FileE), "ALL", None, "INPUT")
print("-----> Der Intersect-Algorithmus wurde erfolgreich ausgefuehrt...")
print("-----> " + OUT_FileoE + IDW_Power + OUT_FileE + " erfolgreich erstellt!")
print("")
print("")

# pressEnterToNext = input("-----> Um Fortzufahren bitte die ENTER-Taste druecken <-----")

# Process: Export Reference-Table To Excel
EW_Excel_Table_Output = "C:\EW-10535-MT\REFERENCE-POINTS\Results_for_IDW_" + DATENSATZ + ".xlsx"
arcpy.TableToExcel_conversion(EWS_Reference_Table, EW_Excel_Table_Output, "NAME", "CODE")

# Process with Kriging as Algorithm/Tool 1
# -----
if InterpolationAuswahl == "2":
# Interim Results Definition
ZE1oE = "C:\EW-10535-MT\OUTPUT-ArcGIS-Pro\ZE1_Kriging_"
ZE1E = ".tif"
ZE2PROJECToE = "C:\EW-10535-MT\OUTPUT-ArcGIS-Pro\ZE2_PROJECT_mit_Kriging_"
ZE2PROJECTE = ".tif"
ZE3CONTOURoE = "C:\EW-10535-MT\OUTPUT-ArcGIS-Pro\ZE3_CONTOUR_mit_Kriging_"
ZE3CONTOURE = ".shp"
OUT_FileoE = "C:\EW-10535-MT\OUTPUT-ArcGIS-Pro\ERGEBNIS_mit_Kriging_"
OUT_FileE = ".shp"
Output_Variance_of_Prediction_Raster = ""

# Time delay
time.sleep(3)

# Clear Screen
os.system("CLS")

print("")
print("")
print("")
print("-----")
print("-----> Bitte warten - Berechnungen werden durchgeführt... <-----")
print("-----")
print("-----> Bitte warten - Berechnungen werden durchgeführt... <-----")
print("-----")
print("-----> Bitte warten - Berechnungen werden durchgeführt... <-----")
print("-----")
print("-----> Bitte warten - Berechnungen werden durchgeführt... <-----")
print("-----")
print("")
print("")
print("")

# Performing Algorithm/Tool 1 - Kriging ---> with Kriging-Type: Ordinary -> Spherical
Kriging_Type = "SPH"
SemiVariogram_Kriging = "Spherical 1,000000 # # #"
arcpy.gp.Kriging_sa(IN_File, Z_Field_Kriging, (ZE1oE + Kriging_Type + ZE1E), SemiVariogram_Kriging, Cell_Size_Kriging, Search_Ra-
dius_Kriging, Output_Variance_of_Prediction_Raster)
print("")
print("")
print("-----> Der Kriging-Algorithmus wurde mit Kriging-Typ = " + Kriging_Type + " erfolgreich ausgefuehrt...")
print("Kriging-Variante: " + Kriging_Type)
print("-----> " + ZE1oE + Kriging_Type + ZE1E + " erfolgreich erstellt!")
print("")
print("")

# -----
# Raster-Properties (Statistics) for Result of Algorithm/Tool 1
# Process: Get Raster Properties ---> MINIMUM
EW_MIN_Result = arcpy.GetRasterProperties_management((ZE1oE + Kriging_Type + ZE1E), property_type="MINIMUM", band_index="Band_1")
EW_MIN = EW_MIN_Result.getOutput(0)
print("MIN-Wert: " + EW_MIN)
print("")
# Process: Get Raster Properties ---> MAXIMUM
EW_MAX_Result = arcpy.GetRasterProperties_management((ZE1oE + Kriging_Type + ZE1E), property_type="MAXIMUM", band_index="Band_1")
EW_MAX = EW_MAX_Result.getOutput(0)
print("MAX-Wert: " + EW_MAX)
print("")

```

```

# Process: Get Raster Properties ----> MEAN
EW_MEAN_Result = arcpy.GetRasterProperties_management((ZE1oE + Kriging_Type + ZE1E), property_type="MEAN", band_index="Band_1")
EW_MEAN = EW_MEAN_Result.getOutput(0)
print("MEAN-Wert: " + EW_MEAN)
print("")
# Process: Get Raster Properties ----> STD-Deviation
EW_STD_Result = arcpy.GetRasterProperties_management((ZE1oE + Kriging_Type + ZE1E), property_type="STD", band_index="Band_1")
EW_STD = EW_STD_Result.getOutput(0)
print("STD-Wert: " + EW_STD)
print("")
# Process: Get Raster Properties ----> noDATA
EW_noDATA_Result = arcpy.GetRasterProperties_management((ZE1oE + Kriging_Type + ZE1E), property_type="ANYNODATA", band_in-
dex="Band_1")
EW_noDATA = EW_noDATA_Result.getOutput(0)
print("Nulls-Werte: " + EW_noDATA)
print("")
# -----
# -----
# Process: Extract Reference-Values To Table
In_Feature_EW = "C:\EW-10535-MT\REFERENCE-POINTS\Referencepoints.shp"
In_Raster_EW = ZE1oE + Kriging_Type + ZE1E
Out_Table_EW = ZE1oE + Kriging_Type + "_Table.dbf"
Out_Raster_Names_Table_EW = ""
arcpy.ExtractValuesToTable_ga(In_Feature_EW, In_Raster_EW, Out_Table_EW, Out_Raster_Names_Table_EW, "ADD_WARNING_FIELD")
print("-----> " + Out_Table_EW + " erfolgreich erstellt!")
print("")
# -----
# Process: Add Fields to Table
EW_field_name_1 = Kriging_Type
EW_field_name_2 = "D_" + Kriging_Type
EW_field_name_3 = "Q_" + Kriging_Type
arcpy.AddField_management(Out_Table_EW, EW_field_name_1, "DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
arcpy.AddField_management(Out_Table_EW, EW_field_name_2, "DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
arcpy.AddField_management(Out_Table_EW, EW_field_name_3, "DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
# Process: Calculate Field in Table
arcpy.CalculateField_management(Out_Table_EW, EW_field_name_1, "!Value!", "PYTHON3", "")
# Process: Delete Fields in Table
arcpy.DeleteField_management(Out_Table_EW, "Value")
arcpy.DeleteField_management(Out_Table_EW, "SrcID_Feat")
arcpy.DeleteField_management(Out_Table_EW, "SrcID_Rast")
# Process: Join Fields in Table and Reference-Table
EWS_Reference_Table = "C:\EW-10535-MT\REFERENCE-TABLES\Referencepoints_Table.dbf"
EWS_Fields = EW_field_name_1 + ";" + EW_field_name_2 + ";" + EW_field_name_3
arcpy.JoinField_management(EWS_Reference_Table, "OID", Out_Table_EW, "OID", EWS_Fields)
# Process: Calculate Field in Reference-Table
EW_Expression_1 = "!" + EW_field_name_1 + "! - !POINT_Z!"
arcpy.CalculateField_management(EWS_Reference_Table, EW_field_name_2, EW_Expression_1, "PYTHON3", "")
# Process: Calculate Field in Reference-Table
EW_Expression_2 = "!" + EW_field_name_2 + "! * !" + EW_field_name_2 + "!"
arcpy.CalculateField_management(EWS_Reference_Table, EW_field_name_3, EW_Expression_2, "PYTHON3", "")
# -----
# Performing Algorithm/Tool 2 - "Project Raster"
arcpy.management.ProjectRaster((ZE1oE + Kriging_Type + ZE1E), (ZE2PROJECToE + Kriging_Type + ZE2PROJECTE), Coordinate_System_Output,
Resampling_Technique, Cell_Size_2, Geographic_Transformation, None, Coordinate_System_Input)
print("-----> Der Project Raster-Algorithmus wurde erfolgreich ausgeführt...")
print("-----> " + ZE2PROJECToE + Kriging_Type + ZE2PROJECTE + " erfolgreich erstellt!")
print("")
print("")
# Performing Algorithm/Tool 3 - "Contour"
arcpy.sa.Contour((ZE2PROJECToE + Kriging_Type + ZE2PROJECTE), (ZE3CONTOURoE + Kriging_Type + ZE3CONTOURE), Contour_Interval,
Base_Contour, Z_Factor)
print("-----> Der Contour-Algorithmus wurde erfolgreich ausgeführt...")
print("-----> " + ZE3CONTOURoE + Kriging_Type + ZE3CONTOURE + " erfolgreich erstellt!")
print("")
print("")
# Performing Algorithm/Tool 4 - "Intersect"
IN_INTERSECT_Tool = [(ZE3CONTOURoE + Kriging_Type + ZE3CONTOURE), Selection_Area]
arcpy.analysis.Intersect(IN_INTERSECT_Tool, (OUT_FileoE + Kriging_Type + OUT_FileE), "ALL", None, "INPUT")
print("-----> Der Intersect-Algorithmus wurde erfolgreich ausgeführt...")
print("-----> " + OUT_FileoE + Kriging_Type + OUT_FileE + " erfolgreich erstellt!")
print("")
print("")
print("")
# pressEnterToExit = input("-----> Um Fortzufahren bitte die ENTER-Taste druecken <-----")

# Reset Kriging_Type-Variable
Kriging_Type = ""
# Reset SemiVariogram_Kriging-Variable
SemiVariogram_Kriging = ""

# Clear Screen
os.system("CLS")

print("")
print("")
print("")
print("-----")
print("-----> Bitte warten - Berechnungen werden durchgeführt... <-----")
print("-----")
print("-----")
print("-----> Bitte warten - Berechnungen werden durchgeführt... <-----")
print("-----")
print("-----")
print("-----> Bitte warten - Berechnungen werden durchgeführt... <-----")
print("-----")
print("-----")

```

```

print("")
print("")
print("")

# Performing Algorithm/Tool 1 - Kriging ----> with Kriging-Type: Ordinary -> Exponential
Kriging_Type = "EXP"
SemiVariogram_Kriging = "Exponential 1,000000 # # #"
arcpy.gp.Kriging_sa(IN_File, Z_Field_Kriging, (ZE1oE + Kriging_Type + ZE1E), SemiVariogram_Kriging, Cell_Size_Kriging, Search_Radius_Kriging, Output_Variance_of_Prediction_Raster)
print("")
print("")
print("-----> Der Kriging-Algorithmus wurde mit Kriging-Typ = " + Kriging_Type + " erfolgreich ausgefuehrt..")
print("Kriging-Variante: " + Kriging_Type)
print("-----> " + ZE1oE + Kriging_Type + ZE1E + " erfolgreich erstellt!")
print("")
print("")

# -----
# Raster-Properties (Statistics) for Result of Algorithm/Tool 1
# Process: Get Raster Properties ----> MINIMUM
EW_MIN_Result = arcpy.GetRasterProperties_management((ZE1oE + Kriging_Type + ZE1E), property_type="MINIMUM", band_index="Band_1")
EW_MIN = EW_MIN_Result.getOutput(0)
print("MIN-Wert: " + EW_MIN)
print("")
# Process: Get Raster Properties ----> MAXIMUM
EW_MAX_Result = arcpy.GetRasterProperties_management((ZE1oE + Kriging_Type + ZE1E), property_type="MAXIMUM", band_index="Band_1")
EW_MAX = EW_MAX_Result.getOutput(0)
print("MAX-Wert: " + EW_MAX)
print("")
# Process: Get Raster Properties ----> MEAN
EW_MEAN_Result = arcpy.GetRasterProperties_management((ZE1oE + Kriging_Type + ZE1E), property_type="MEAN", band_index="Band_1")
EW_MEAN = EW_MEAN_Result.getOutput(0)
print("MEAN-Wert: " + EW_MEAN)
print("")
# Process: Get Raster Properties ----> STD-Deviation
EW_STD_Result = arcpy.GetRasterProperties_management((ZE1oE + Kriging_Type + ZE1E), property_type="STD", band_index="Band_1")
EW_STD = EW_STD_Result.getOutput(0)
print("STD-Wert: " + EW_STD)
print("")
# Process: Get Raster Properties ----> noDATA
EW_noDATA_Result = arcpy.GetRasterProperties_management((ZE1oE + Kriging_Type + ZE1E), property_type="ANYNODATA", band_index="Band_1")
EW_noDATA = EW_noDATA_Result.getOutput(0)
print("Nulls-Werte: " + EW_noDATA)
print("")
# -----

# Process: Extract Reference-Values To Table
In_Feature_EW = "C:\EW-10535-MT\REFERENCE-POINTS\Referencepoints.shp"
In_Raster_EW = ZE1oE + Kriging_Type + ZE1E
Out_Table_EW = ZE1oE + Kriging_Type + "_Table.dbf"
Out_Raster_Names_Table_EW = ""
arcpy.ExtractValuesToTable_ga(In_Feature_EW, In_Raster_EW, Out_Table_EW, Out_Raster_Names_Table_EW, "ADD_WARNING_FIELD")
print("-----> " + Out_Table_EW + " erfolgreich erstellt!")
print("")
# -----

# Process: Add Fields to Table
EW_field_name_1 = Kriging_Type
EW_field_name_2 = "D_" + Kriging_Type
EW_field_name_3 = "Q_" + Kriging_Type
arcpy.AddField_management(Out_Table_EW, EW_field_name_1, "DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
arcpy.AddField_management(Out_Table_EW, EW_field_name_2, "DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
arcpy.AddField_management(Out_Table_EW, EW_field_name_3, "DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
# Process: Calculate Field in Table
arcpy.CalculateField_management(Out_Table_EW, EW_field_name_1, "!Value!", "PYTHON3", "")
# Process: Delete Fields in Table
arcpy.DeleteField_management(Out_Table_EW, "Value")
arcpy.DeleteField_management(Out_Table_EW, "SrcID_Feat")
arcpy.DeleteField_management(Out_Table_EW, "SrcID_Rast")
# Process: Join Fields in Table and Reference-Table
EWS_Reference_Table = "C:\EW-10535-MT\REFERENCE-TABLES\Referencepoints_Table.dbf"
EWS_Fields = EW_field_name_1 + ";" + EW_field_name_2 + ";" + EW_field_name_3
arcpy.JoinField_management(EWS_Reference_Table, "OID", Out_Table_EW, "OID", EWS_Fields)
# Process: Calculate Field in Reference-Table
EW_Expression_1 = "! " + EW_field_name_1 + "! - !POINT_Z!"
arcpy.CalculateField_management(EWS_Reference_Table, EW_field_name_2, EW_Expression_1, "PYTHON3", "")
# Process: Calculate Field in Reference-Table
EW_Expression_2 = "! " + EW_field_name_2 + "! * !" + EW_field_name_2 + "! "
arcpy.CalculateField_management(EWS_Reference_Table, EW_field_name_3, EW_Expression_2, "PYTHON3", "")
# -----

# Performing Algorithm/Tool 2 - "Project Raster"
arcpy.management.ProjectRaster((ZE1oE + Kriging_Type + ZE1E), (ZE2PROJECToE + Kriging_Type + ZE2PROJECTE), Coordinate_System_Output, Resampling_Technique, Cell_Size_2, Geographic_Transformation, None, Coordinate_System_Input)
print("-----> Der Project Raster-Algorithmus wurde erfolgreich ausgefuehrt..")
print("-----> " + ZE2PROJECToE + Kriging_Type + ZE2PROJECTE + " erfolgreich erstellt!")
print("")
print("")

# Performing Algorithm/Tool 3 - "Contour"
arcpy.sa.Contour((ZE2PROJECToE + Kriging_Type + ZE2PROJECTE), (ZE3CONTOURoE + Kriging_Type + ZE3CONTOURE), Contour_Interval, Base_Contour, Z_Factor)
print("-----> Der Contour-Algorithmus wurde erfolgreich ausgefuehrt..")
print("-----> " + ZE3CONTOURoE + Kriging_Type + ZE3CONTOURE + " erfolgreich erstellt!")
print("")
print("")

# Performing Algorithm/Tool 4 - "Intersect"
IN_INTERSECT_Tool = [(ZE3CONTOURoE + Kriging_Type + ZE3CONTOURE), Selection_Area]
arcpy.analysis.Intersect(IN_INTERSECT_Tool, (OUT_FileoE + Kriging_Type + OUT_FileE), "ALL", None, "INPUT")
print("-----> Der Intersect-Algorithmus wurde erfolgreich ausgefuehrt..")

```



```

arcpy.JoinField_management(EWS_Reference_Table, "OID", Out_Table_EW, "OID", EWS_Fields)
# Process: Calculate Field in Reference-Table
EW_Expression_1 = "!" + EW_field_name_1 + "! - !POINT_Z!"
arcpy.CalculateField_management(EWS_Reference_Table, EW_field_name_2, EW_Expression_1, "PYTHON3", "")
# Process: Calculate Field in Reference-Table
EW_Expression_2 = "!" + EW_field_name_2 + "! * !" + EW_field_name_2 + "!"
arcpy.CalculateField_management(EWS_Reference_Table, EW_field_name_3, EW_Expression_2, "PYTHON3", "")
# -----

# Performing Algorithm/Tool 2 - "Project Raster"
arcpy.management.ProjectRaster((ZE1oE + Kriging_Type + ZE1E), (ZE2PROJECToE + Kriging_Type + ZE2PROJECTE), Coordinate_System_Output,
Resampling_Technique, Cell_Size_2, Geographic_Transformation, None, Coordinate_System_Input)
print("-----> Der Project Raster-Algorithmus wurde erfolgreich ausgefuehrt...")
print("-----> " + ZE2PROJECToE + Kriging_Type + ZE2PROJECTE + " erfolgreich erstellt!")
print("")
print("")

# Performing Algorithm/Tool 3 - "Contour"
arcpy.sa.Contour((ZE2PROJECToE + Kriging_Type + ZE2PROJECTE), (ZE3CONTOURoE + Kriging_Type + ZE3CONTOURE), Contour_Interval,
Base_Contour, Z_Factor)
print("-----> Der Contour-Algorithmus wurde erfolgreich ausgefuehrt...")
print("-----> " + ZE3CONTOURoE + Kriging_Type + ZE3CONTOURE + " erfolgreich erstellt!")
print("")
print("")

# Performing Algorithm/Tool 4 - "Intersect"
IN_INTERSECT_Tool = [(ZE3CONTOURoE + Kriging_Type + ZE3CONTOURE), Selection_Area]
arcpy.analysis.Intersect(IN_INTERSECT_Tool, (OUT_FileoE + Kriging_Type + OUT_FileE), "ALL", None, "INPUT")
print("-----> Der Intersect-Algorithmus wurde erfolgreich ausgefuehrt...")
print("-----> " + OUT_FileoE + Kriging_Type + OUT_FileE + " erfolgreich erstellt!")
print("")
print("")
print("")

# pressEnterToExit = input("-----> Um Fortzufahren bitte die ENTER-Taste druecken <-----")

# Reset Kriging_Type-Variable
Kriging_Type = ""
# Reset SemiVariogram_Kriging-Variable
SemiVariogram_Kriging = ""

# Clear Screen
os.system("CLS")

print("")
print("")
print("")
print("-----")
print("-----> Bitte warten - Berechnungen werden durchgefuehrt... <-----")
print("-----")
print("-----> Bitte warten - Berechnungen werden durchgefuehrt... <-----")
print("-----")
print("-----> Bitte warten - Berechnungen werden durchgefuehrt... <-----")
print("-----")
print("-----> Bitte warten - Berechnungen werden durchgefuehrt... <-----")
print("-----")
print("-----")
print("-----")
print("-----")

# Performing Algorithm/Tool 1 - Kriging ----> with Kriging-Type: Universal -> Linear Drift
Kriging_Type = "LiDr"
SemiVariogram_Kriging = "LinearDrift 1,000000 # # #"
arcpy.gp.Kriging_sa(IN_File, Z_Field_Kriging, (ZE1oE + Kriging_Type + ZE1E), SemiVariogram_Kriging, Cell_Size_Kriging, Search_Radius_Kriging, Output_Variance_of_Prediction_Raster)
print("")
print("")
print("-----> Der Kriging-Algorithmus wurde mit Kriging-Typ = " + Kriging_Type + " erfolgreich ausgefuehrt...")
print("Kriging-Variante: " + Kriging_Type)
print("-----> " + ZE1oE + Kriging_Type + ZE1E + " erfolgreich erstellt!")
print("")
print("")

# -----
# Raster-Properties (Statistics) for Result of Algorithm/Tool 1
# Process: Get Raster Properties ----> MINIMUM
EW_MIN_Result = arcpy.GetRasterProperties_management((ZE1oE + Kriging_Type + ZE1E), property_type="MINIMUM", band_index="Band_1")
EW_MIN = EW_MIN_Result.getOutput(0)
print("MIN-Wert: " + EW_MIN)
print("")
# Process: Get Raster Properties ----> MAXIMUM
EW_MAX_Result = arcpy.GetRasterProperties_management((ZE1oE + Kriging_Type + ZE1E), property_type="MAXIMUM", band_index="Band_1")
EW_MAX = EW_MAX_Result.getOutput(0)
print("MAX-Wert: " + EW_MAX)
print("")
# Process: Get Raster Properties ----> MEAN
EW_MEAN_Result = arcpy.GetRasterProperties_management((ZE1oE + Kriging_Type + ZE1E), property_type="MEAN", band_index="Band_1")
EW_MEAN = EW_MEAN_Result.getOutput(0)
print("MEAN-Wert: " + EW_MEAN)
print("")
# Process: Get Raster Properties ----> STD-Deviation
EW_STD_Result = arcpy.GetRasterProperties_management((ZE1oE + Kriging_Type + ZE1E), property_type="STD", band_index="Band_1")
EW_STD = EW_STD_Result.getOutput(0)
print("STD-Wert: " + EW_STD)
print("")
# Process: Get Raster Properties ----> noDATA
EW_noDATA_Result = arcpy.GetRasterProperties_management((ZE1oE + Kriging_Type + ZE1E), property_type="ANYNODATA", band_index="Band_1")
EW_noDATA = EW_noDATA_Result.getOutput(0)
print("Nulls-Werte: " + EW_noDATA)

```

```

print("")
# -----
# -----
# Process: Extract Reference-Values To Table
In_Feature_EW = "C:\EW-10535-MT\REFERENCE-POINTS\Referencepoints.shp"
In_Raster_EW = ZE1oE + Kriging_Type + ZE1E
Out_Table_EW = ZE1oE + Kriging_Type + "_Table.dbf"
Out_Raster_Names_Table_EW = ""
arcpy.ExtractValuesToTable_ga(In_Feature_EW, In_Raster_EW, Out_Table_EW, Out_Raster_Names_Table_EW, "ADD_WARNING_FIELD")
print("-----> " + Out_Table_EW + " erfolgreich erstellt!")
print("")
# -----
# Process: Add Fields to Table
EW_field_name_1 = Kriging_Type
EW_field_name_2 = "D_" + Kriging_Type
EW_field_name_3 = "Q_" + Kriging_Type
arcpy.AddField_management(Out_Table_EW, EW_field_name_1, "DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
arcpy.AddField_management(Out_Table_EW, EW_field_name_2, "DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
arcpy.AddField_management(Out_Table_EW, EW_field_name_3, "DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
# Process: Calculate Field in Table
arcpy.CalculateField_management(Out_Table_EW, EW_field_name_1, "!Value!", "PYTHON3", "")
# Process: Delete Fields in Table
arcpy.DeleteField_management(Out_Table_EW, "Value")
arcpy.DeleteField_management(Out_Table_EW, "SrcID_Feat")
arcpy.DeleteField_management(Out_Table_EW, "SrcID_Rast")
# Process: Join Fields in Table and Reference-Table
EWS_Reference_Table = "C:\EW-10535-MT\REFERENCE-TABLES\Referencepoints_Table.dbf"
EWS_Fields = EW_field_name_1 + ";" + EW_field_name_2 + ";" + EW_field_name_3
arcpy.JoinField_management(EWS_Reference_Table, "OID", Out_Table_EW, "OID", EWS_Fields)
# Process: Calculate Field in Reference-Table
EW_Expression_1 = "!" + EW_field_name_1 + "! - !POINT Z!"
arcpy.CalculateField_management(EWS_Reference_Table, EW_field_name_2, EW_Expression_1, "PYTHON3", "")
# Process: Calculate Field in Reference-Table
EW_Expression_2 = "!" + EW_field_name_2 + "! * !" + EW_field_name_2 + "!"
arcpy.CalculateField_management(EWS_Reference_Table, EW_field_name_3, EW_Expression_2, "PYTHON3", "")
# -----

# Performing Algorithm/Tool 2 - "Project Raster"
arcpy.management.ProjectRaster((ZE1oE + Kriging_Type + ZE1E), (ZE2PROJECToE + Kriging_Type + ZE2PROJECTE), Coordinate_System_Output,
Resampling_Technique, Cell_Size_2, Geographic_Transformation, None, Coordinate_System_Input)
print("-----> Der Project Raster-Algorithmus wurde erfolgreich ausgefuehrt...")
print("-----> " + ZE2PROJECToE + Kriging_Type + ZE2PROJECTE + " erfolgreich erstellt!")
print("")
print("")

# Performing Algorithm/Tool 3 - "Contour"
arcpy.sa.Contour((ZE2PROJECToE + Kriging_Type + ZE2PROJECTE), (ZE3CONTOURoE + Kriging_Type + ZE3CONTOURE), Contour_Interval,
Base_Contour, Z_Factor)
print("-----> Der Contour-Algorithmus wurde erfolgreich ausgefuehrt...")
print("-----> " + ZE3CONTOURoE + Kriging_Type + ZE3CONTOURE + " erfolgreich erstellt!")
print("")
print("")

# Performing Algorithm/Tool 4 - "Intersect"
IN_INTERSECT_Tool = [(ZE3CONTOURoE + Kriging_Type + ZE3CONTOURE), Selection_Area]
arcpy.analysis.Intersect(IN_INTERSECT_Tool, (OUT_FileoE + Kriging_Type + OUT_FileE), "ALL", None, "INPUT")
print("-----> Der Intersect-Algorithmus wurde erfolgreich ausgefuehrt...")
print("-----> " + OUT_FileoE + Kriging_Type + OUT_FileE + " erfolgreich erstellt!")
print("")
print("")
print("")

# pressEnterToExit = input("-----> Um Fortzufahren bitte die ENTER-Taste druecken <-----")

# Reset Kriging_Type-Variable
Kriging_Type = ""
# Reset SemiVariogram_Kriging-Variable
SemiVariogram_Kriging = ""

# Process: Export Reference-Table To Excel
EW_Excel_Table_Output = "C:\EW-10535-MT\REFERENCE-POINTS\Results_for_Kriging_" + DATENSATZ + ".xlsx"
arcpy.TableToExcel_conversion(EWS_Reference_Table, EW_Excel_Table_Output, "NAME", "CODE")

# Process with Spline as Algorithm/Tool 1
# -----
if InterpolationAuswahl == "3":
# Interim Results Definition
ZE1oE = "C:\EW-10535-MT\OUTPUT-ArcGIS-Pro\ZE1_Spline-Weight_"
ZE1E = ".tif"
ZE2PROJECToE = "C:\EW-10535-MT\OUTPUT-ArcGIS-Pro\ZE2_PROJECT_mit_Spline-Weight_"
ZE2PROJECTE = ".tif"
ZE3CONTOURoE = "C:\EW-10535-MT\OUTPUT-ArcGIS-Pro\ZE3_CONTOUR_mit_Spline-Weight_"
ZE3CONTOURE = ".shp"
OUT_FileoE = "C:\EW-10535-MT\OUTPUT-ArcGIS-Pro\ERGBNIS_mit_Spline-Weight_"
OUT_FileE = ".shp"

SPLINE_Weight_Bereich = ["0,05", "0,1", "0,15", "0,2", "0,25", "0,3", "0,35", "0,4", "0,45", "0,5"]
for SPLINE_Weight in SPLINE_Weight_Bereich:

# Time delay
time.sleep(3)
# Clear Screen
os.system("CLS")
print("")
print("")
print("")
print("-----")
print("-----> Bitte warten - Berechnungen werden durchgefuehrt.... <-----")
print("-----")
print("-----")

```

```

print("-----> Bitte warten - Berechnungen werden durchgeführt... <-----")
print("-----")
print("-----")
print("-----> Bitte warten - Berechnungen werden durchgeführt... <-----")
print("-----")
print("-----")
print("-----")
print("-----")

# Performing Algorithm/Tool 1 - "Spline"
arcpy.gp.Spline_sa(IN_File, SPLINE_Z_Value_Field, (ZE1oE + SPLINE_Weight + ZE1E), SPLINE_Output_Cell_Size, SPLINE_Type,
SPLINE_Weight, SPLINE_Number_of_Points)
print("-----> Der Spline-Algorithmus wurde mit Weight-Parameter = " + SPLINE_Weight + " erfolgreich ausgeführt...")
print("Weight-Parameter: " + SPLINE_Weight)
print("-----> " + ZE1oE + SPLINE_Weight + ZE1E + " erfolgreich erstellt!")
print("")
print("")

# -----
---
# Raster-Properties (Statistics) for Result of Algorithm/Tool 1
# Process: Get Raster Properties ---> MINIMUM
EW_MIN_Result = arcpy.GetRasterProperties_management((ZE1oE + SPLINE_Weight + ZE1E), property_type="MINIMUM", band_in-
dex="Band_1")
EW_MIN = EW_MIN_Result.getOutput(0)
print("MIN-Wert: " + EW_MIN)
print("")
# Process: Get Raster Properties ---> MAXIMUM
EW_MAX_Result = arcpy.GetRasterProperties_management((ZE1oE + SPLINE_Weight + ZE1E), property_type="MAXIMUM", band_in-
dex="Band_1")
EW_MAX = EW_MAX_Result.getOutput(0)
print("MAX-Wert: " + EW_MAX)
print("")
# Process: Get Raster Properties ---> MEAN
EW_MEAN_Result = arcpy.GetRasterProperties_management((ZE1oE + SPLINE_Weight + ZE1E), property_type="MEAN", band_index="Band_1")
EW_MEAN = EW_MEAN_Result.getOutput(0)
print("MEAN-Wert: " + EW_MEAN)
print("")
# Process: Get Raster Properties ---> STD-Deviation
EW_STD_Result = arcpy.GetRasterProperties_management((ZE1oE + SPLINE_Weight + ZE1E), property_type="STD", band_index="Band_1")
EW_STD = EW_STD_Result.getOutput(0)
print("STD-Wert: " + EW_STD)
print("")
# Process: Get Raster Properties ---> noDATA
EW_noDATA_Result = arcpy.GetRasterProperties_management((ZE1oE + SPLINE_Weight + ZE1E), property_type="ANYNODATA", band_in-
dex="Band_1")
EW_noDATA = EW_noDATA_Result.getOutput(0)
print("Nulls-Werte: " + EW_noDATA)
print("")
# -----
---
# -----
---
# Process: Extract Reference-Values To Table
In_Feature_EW = "C:\EW-10535-MT\REFERENCE-POINTS\Referencepoints.shp"
In_Raster_EW = ZE1oE + SPLINE_Weight + ZE1E
Out_Table_EW = ZE1oE + SPLINE_Weight + "_Table.dbf"
Out_Raster_Names_Table_EW = ""
arcpy.ExtractValuesToTable_ga(In_Feature_EW, In_Raster_EW, Out_Table_EW, Out_Raster_Names_Table_EW, "ADD_WARNING_FIELD")
print("-----> " + Out_Table_EW + " erfolgreich erstellt!")
print("")
# -----
---
# Process: Add Fields to Table
Attribute_Table_1 = float(SPLINE_Weight.replace(",",".")) * 100
Attribute_Table_2 = int(Attribute_Table_1)
Attribute_Table_3 = str(Attribute_Table_2)
EW_field_name_1 = "SPL_W" + Attribute_Table_3
EW_field_name_2 = "D_SPL_W" + Attribute_Table_3
EW_field_name_3 = "Q_SPL_W" + Attribute_Table_3
arcpy.AddField_management(Out_Table_EW, EW_field_name_1, "DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
arcpy.AddField_management(Out_Table_EW, EW_field_name_2, "DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
arcpy.AddField_management(Out_Table_EW, EW_field_name_3, "DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
# Process: Calculate Field in Table
arcpy.CalculateField_management(Out_Table_EW, EW_field_name_1, "!Value!", "PYTHON3", "")
# Process: Delete Fields in Table
arcpy.DeleteField_management(Out_Table_EW, "Value")
arcpy.DeleteField_management(Out_Table_EW, "SrcID_Feat")
arcpy.DeleteField_management(Out_Table_EW, "SrcID_Rast")
# Process: Join Fields in Table and Reference-Table
EWS_Reference_Table = "C:\EW-10535-MT\REFERENCE-TABLES\Referencepoints_Table.dbf"
EWS_Fields = EW_field_name_1 + ";" + EW_field_name_2 + ";" + EW_field_name_3
arcpy.JoinField_management(EWS_Reference_Table, "OID", Out_Table_EW, "OID", EWS_Fields)
# Process: Calculate Field in Reference-Table
EW_Expression_1 = "!" + EW_field_name_1 + "! - !POINT_Z!"
arcpy.CalculateField_management(EWS_Reference_Table, EW_field_name_2, EW_Expression_1, "PYTHON3", "")
# Process: Calculate Field in Reference-Table
EW_Expression_2 = "!" + EW_field_name_2 + "! * !" + EW_field_name_2 + "!"
arcpy.CalculateField_management(EWS_Reference_Table, EW_field_name_3, EW_Expression_2, "PYTHON3", "")
# -----
---
# Performing Algorithm/Tool 2 - "Project Raster"
arcpy.management.ProjectRaster((ZE1oE + SPLINE_Weight + ZE1E), (ZE2PROJECToE + SPLINE_Weight + ZE2PROJECTE), Coordinate_Sys-
tem_Output, Resampling_Technique, Cell_Size_2, Geographic_Transformation, None, Coordinate_System_Input)
print("-----> Der Project Raster-Algorithmus wurde erfolgreich ausgeführt...")
print("-----> " + ZE2PROJECToE + SPLINE_Weight + ZE2PROJECTE + " erfolgreich erstellt!")
print("")
print("")

```

```

# -----
---
# Raster-Properties (Statistics) for Result of Algorithm/Tool 2
# Process: Get Raster Properties ---> MINIMUM
EW_MIN_Result = arcpy.GetRasterProperties_management((ZE2PROJECToE + SPLINE_Weight + ZE2PROJECTE), property_type="MINIMUM",
band_index="Band_1")
EW_MIN = EW_MIN_Result.getOutput(0)
print("MIN-Wert: " + EW_MIN)
print("")
# Process: Get Raster Properties ---> MAXIMUM
EW_MAX_Result = arcpy.GetRasterProperties_management((ZE2PROJECToE + SPLINE_Weight + ZE2PROJECTE), property_type="MAXIMUM",
band_index="Band_1")
EW_MAX = EW_MAX_Result.getOutput(0)
print("MAX-Wert: " + EW_MAX)
print("")
# Process: Get Raster Properties ---> MEAN
EW_MEAN_Result = arcpy.GetRasterProperties_management((ZE2PROJECToE + SPLINE_Weight + ZE2PROJECTE), property_type="MEAN",
band_index="Band_1")
EW_MEAN = EW_MEAN_Result.getOutput(0)
print("MEAN-Wert: " + EW_MEAN)
print("")
# Process: Get Raster Properties ---> STD-Deviation
EW_STD_Result = arcpy.GetRasterProperties_management((ZE2PROJECToE + SPLINE_Weight + ZE2PROJECTE), property_type="STD", band_in-
dex="Band_1")
EW_STD = EW_STD_Result.getOutput(0)
print("STD-Wert: " + EW_STD)
print("")
# Process: Get Raster Properties ---> noDATA
EW_noDATA_Result = arcpy.GetRasterProperties_management((ZE2PROJECToE + SPLINE_Weight + ZE2PROJECTE), property_type="ANYNODATA",
band_index="Band_1")
EW_noDATA = EW_noDATA_Result.getOutput(0)
print("Nulls-Werte: " + EW_noDATA)
print("")
# -----
---

# Performing Algorithm/Tool 3 - "Contour"
arcpy.sa.Contour((ZE2PROJECToE + SPLINE_Weight + ZE2PROJECTE), (ZE3CONTOURoE + SPLINE_Weight + ZE3CONTOURE), Contour_Interval,
Base_Contour, Z_Factor)
print("-----> Der Contour-Algorithmus wurde erfolgreich ausgefuehrt...")
print("-----> " + ZE3CONTOURoE + SPLINE_Weight + ZE3CONTOURE + " erfolgreich erstellt!")
print("")
print("")

# Performing Algorithm/Tool 4 - "Intersect"
IN_INTERSECT_Tool = [(ZE3CONTOURoE + SPLINE_Weight + ZE3CONTOURE), Selection_Area]
arcpy.analysis.Intersect(IN_INTERSECT_Tool, (OUT_FileoE + SPLINE_Weight + OUT_FileE), "ALL", None, "INPUT")
print("-----> Der Intersect-Algorithmus wurde erfolgreich ausgefuehrt...")
print("-----> " + OUT_FileoE + SPLINE_Weight + OUT_FileE + " erfolgreich erstellt!")
print("")
print("")

# pressEnterToNext = input("-----> Um Fortzufahren bitte die ENTER-Taste druecken <-----")

# Process: Export Reference-Table To Excel
EW_Excel_Table_Output = "C:\EW-10535-MT\REFERENCE-POINTS\Results_for_Spline_" + DATENSATZ + ".xlsx"
arcpy.TableToExcel_conversion(EWS_Reference_Table, EW_Excel_Table_Output, "NAME", "CODE")

print("")
print("")
print("-----")
print("-----> Das Python-Script wurde erfolgreich ausgefuehrt! <-----")
print("-----")
print("-----")

# Time delay
time.sleep(1)

# End of Python-Script

```

## Anhang B – Häufigkeitsverteilungen der Interpolationsergebnisse

- „Inverse Distance Weighted“-Ergebnis mit „Power“-Parameter = 2 (Datensatz 1)

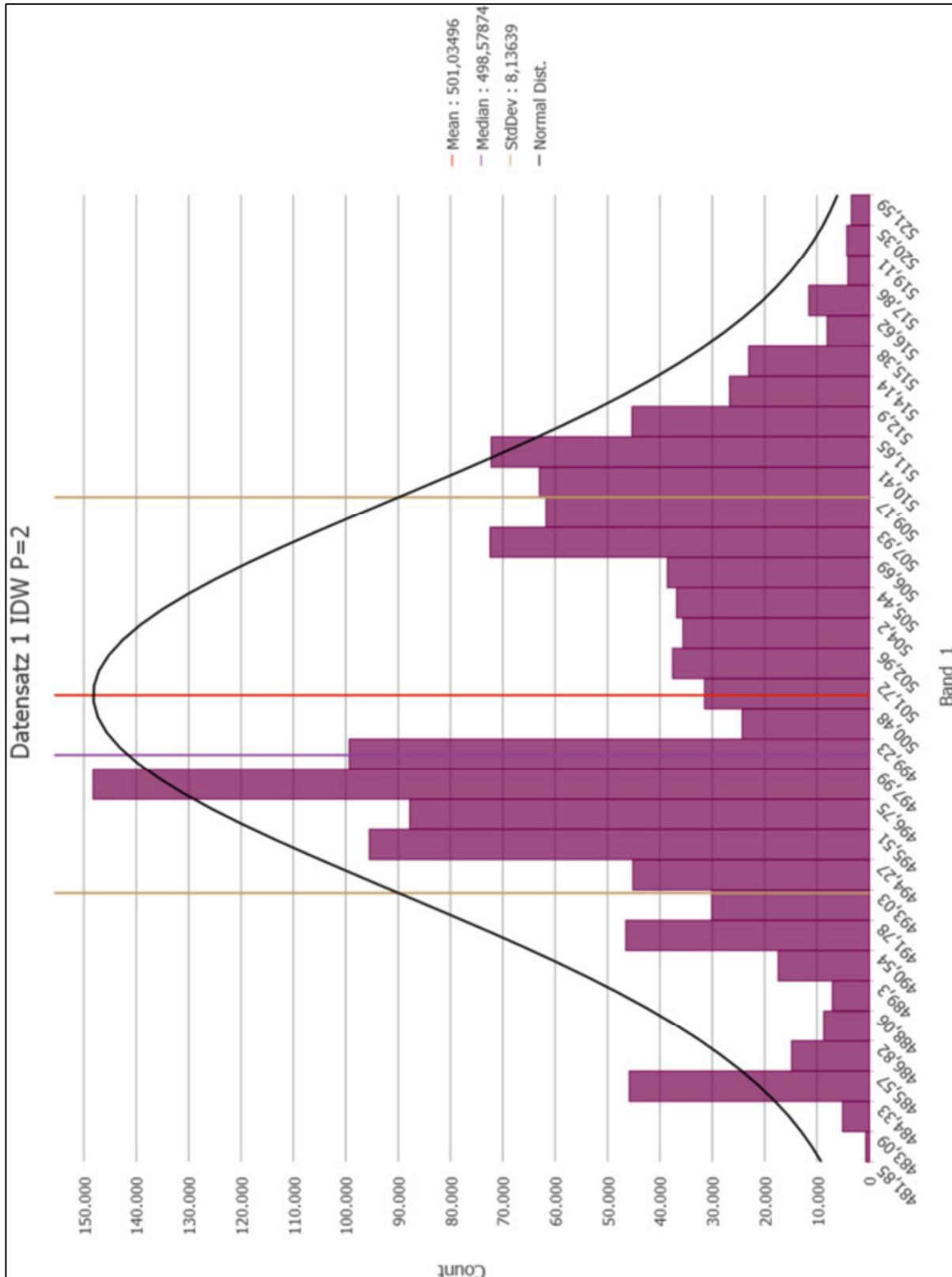


Abbildung 59: Werteverteilung für „IDW“ („Power“ = 2) – Basisdatensatz 1

- „Ordinary Kriging“-Ergebnis mit Semivariogramm = „Exponential“ (Datensatz 1)

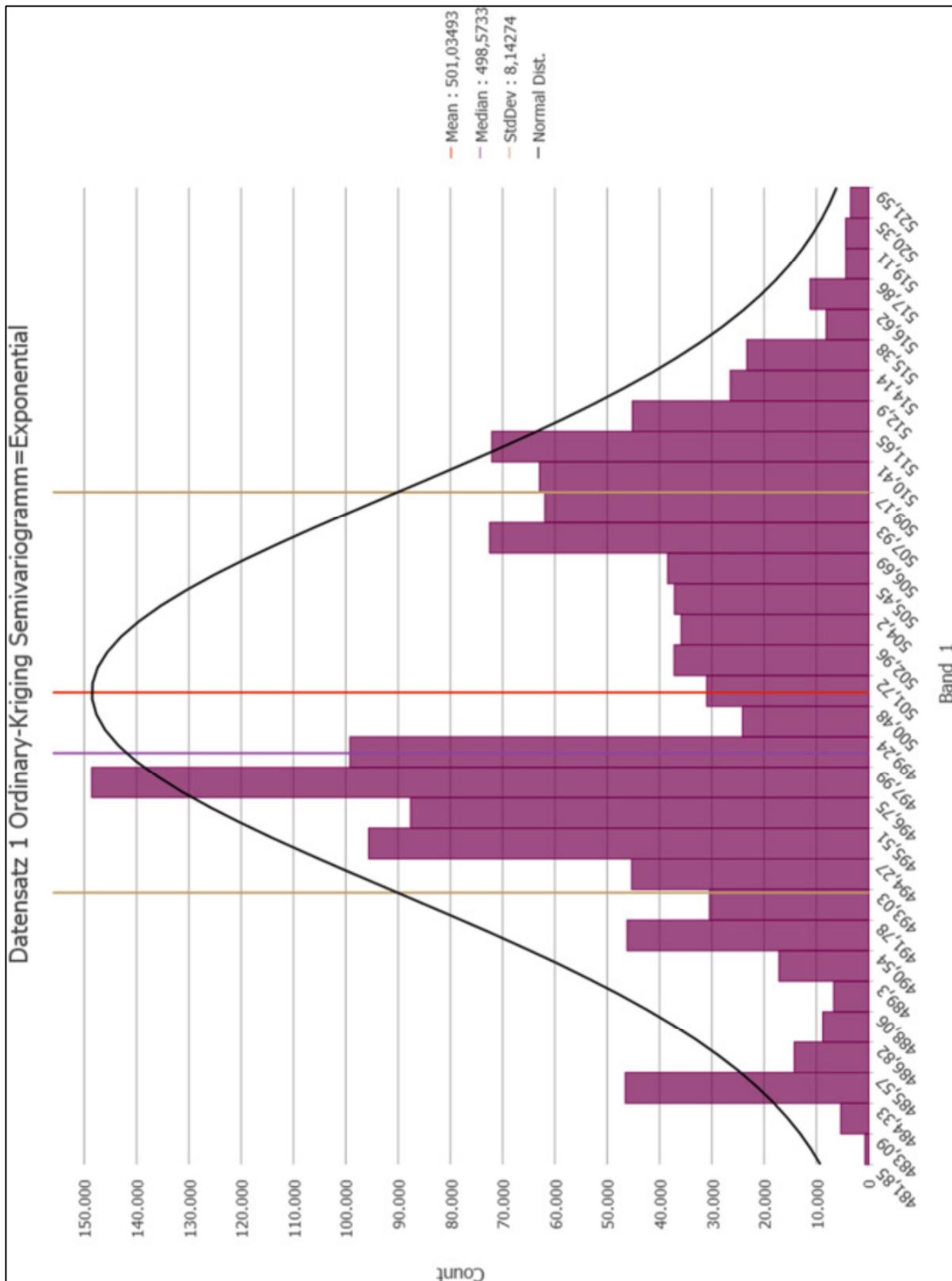


Abbildung 60: Werteverteilung für „Ordinary-Kriging“ („Expon.“) – Basisdatensatz 1

- „Spline“-Ergebnis mit „Weight“-Parameter = 0,1 (Datensatz 1)

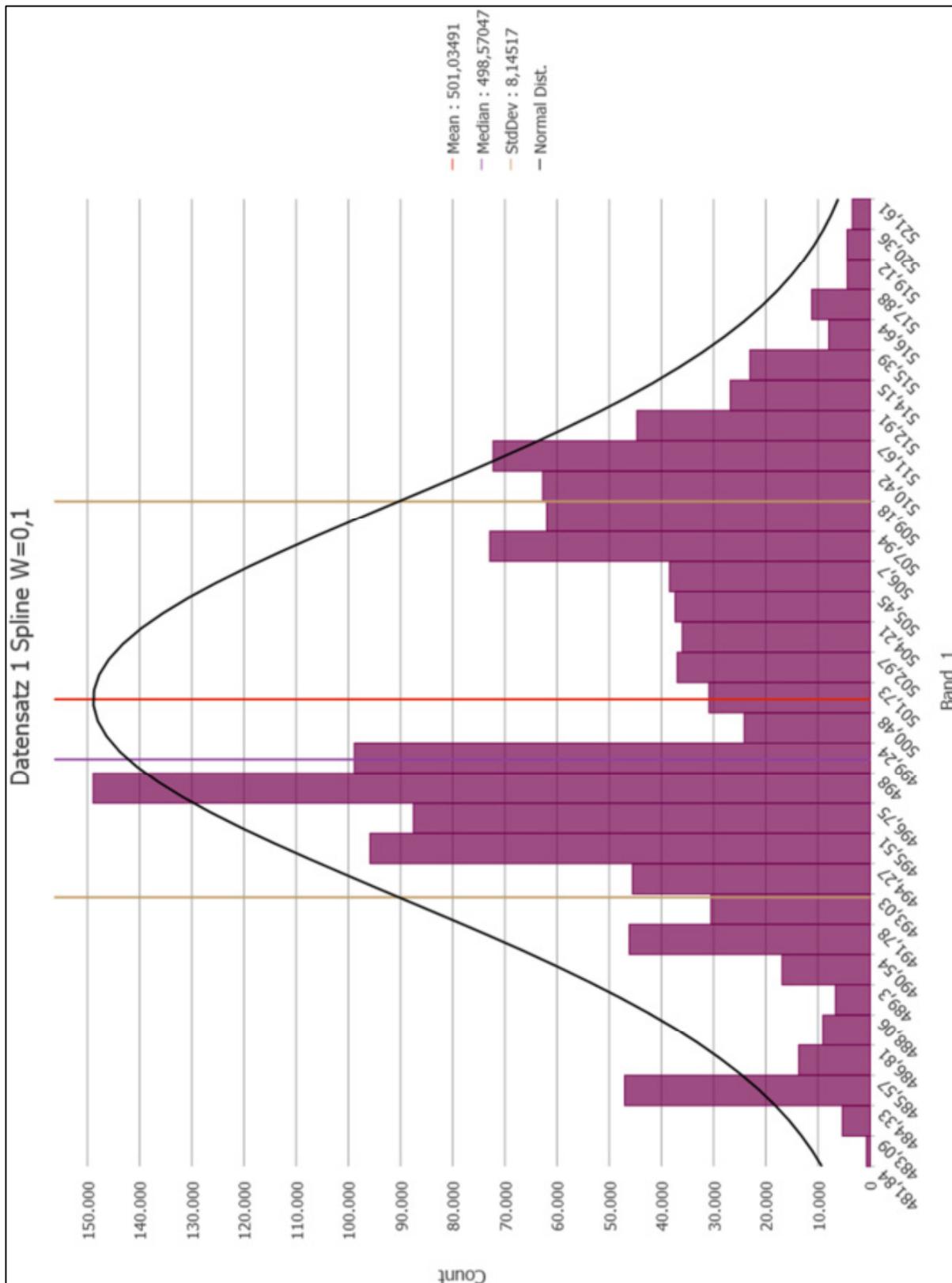


Abbildung 61: Werteverteilung für „Spline“ („Weight“ = 0,1) – Basisdatensatz 1

- „Inverse Distance Weighted“-Ergebnis mit „Power“-Parameter = 3 (Datensatz 2)

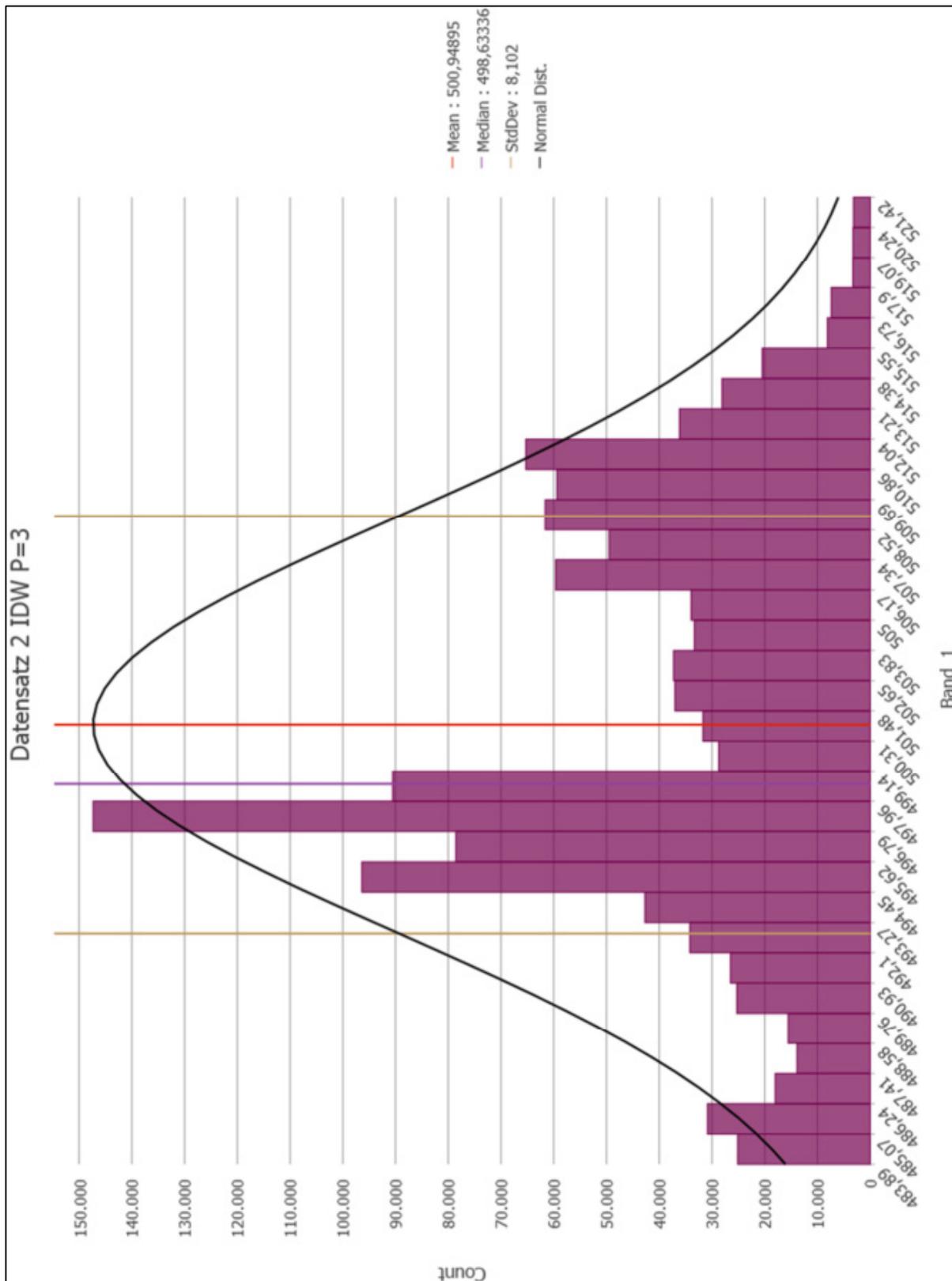


Abbildung 62: Werteverteilung für „IDW“ („Power“ = 3) – Basisdatensatz 2

- „Ordinary Kriging“-Ergebnis mit Semivariogramm = „Exponential“ (Datensatz 2)

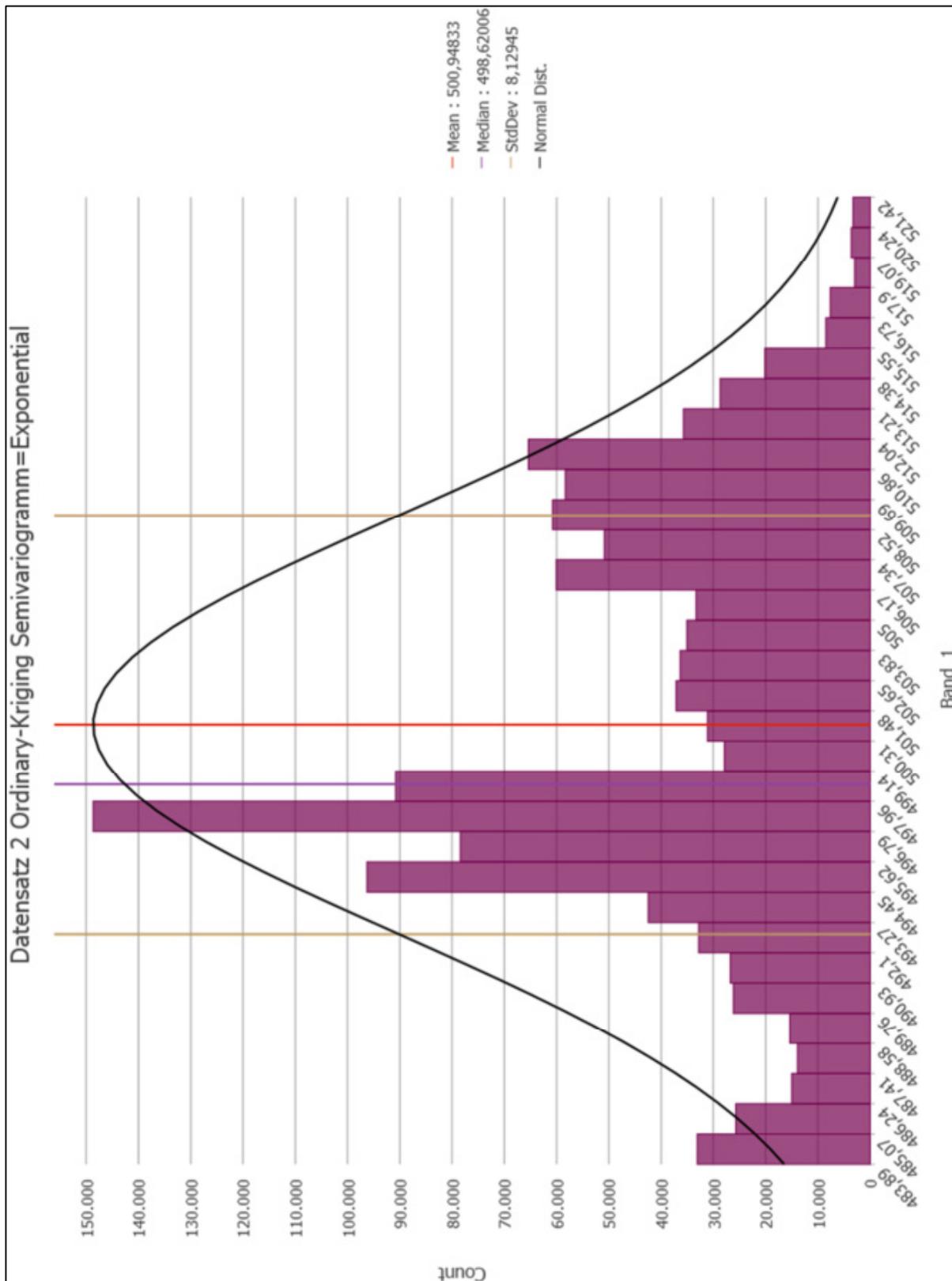


Abbildung 63: Werteverteilung für „Ordinary-Kriging“ („Expon.“) – Basisdatensatz 2

- „Spline“-Ergebnis mit „Weight“-Parameter = 0,05 (Datensatz 2)

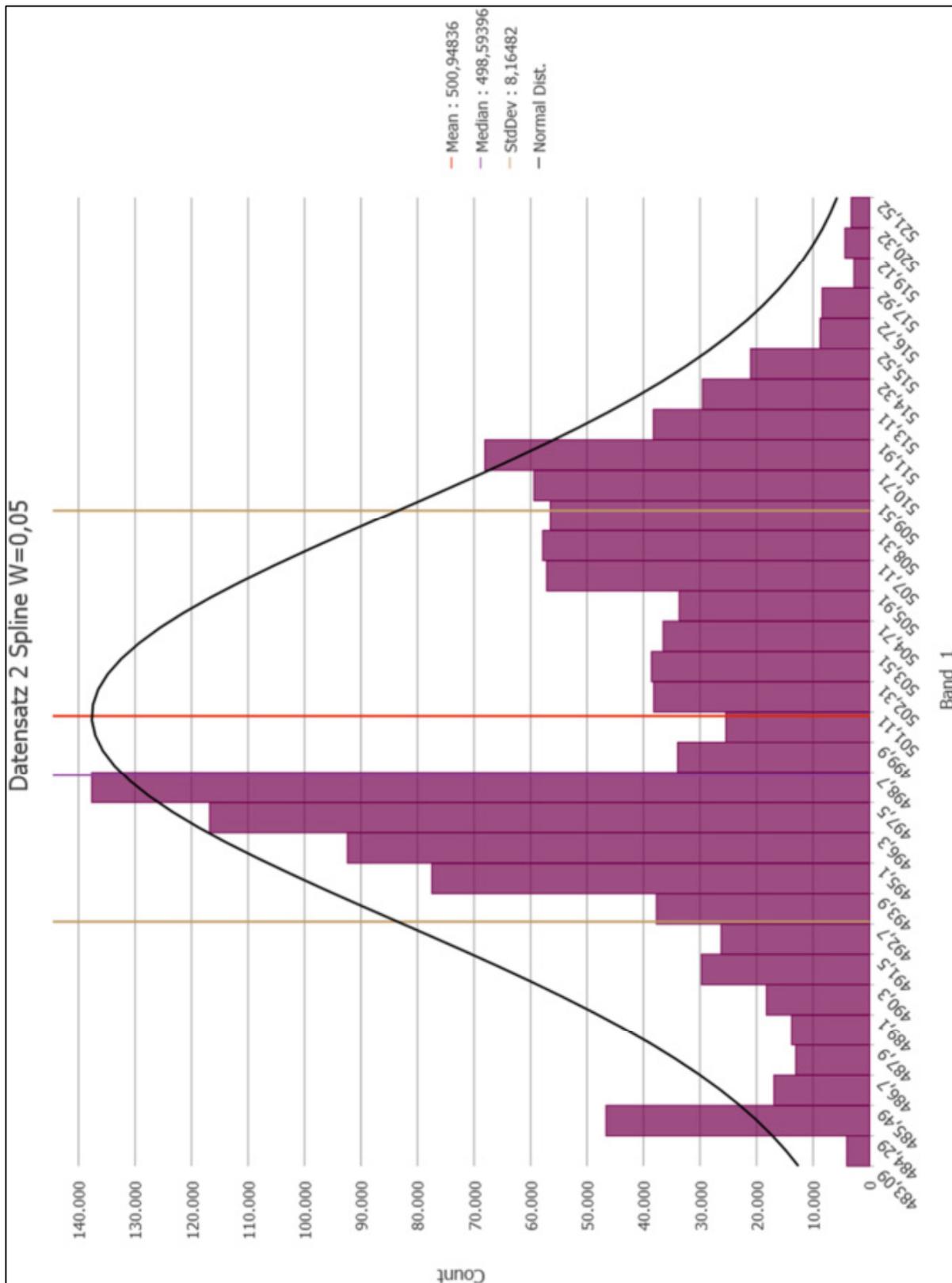


Abbildung 64: Werteverteilung für „Spline“ („Weight“ = 0,05) – Basisdatensatz 2

- „Inverse Distance Weighted“-Ergebnis mit „Power“-Parameter = 3 (Datensatz 3)

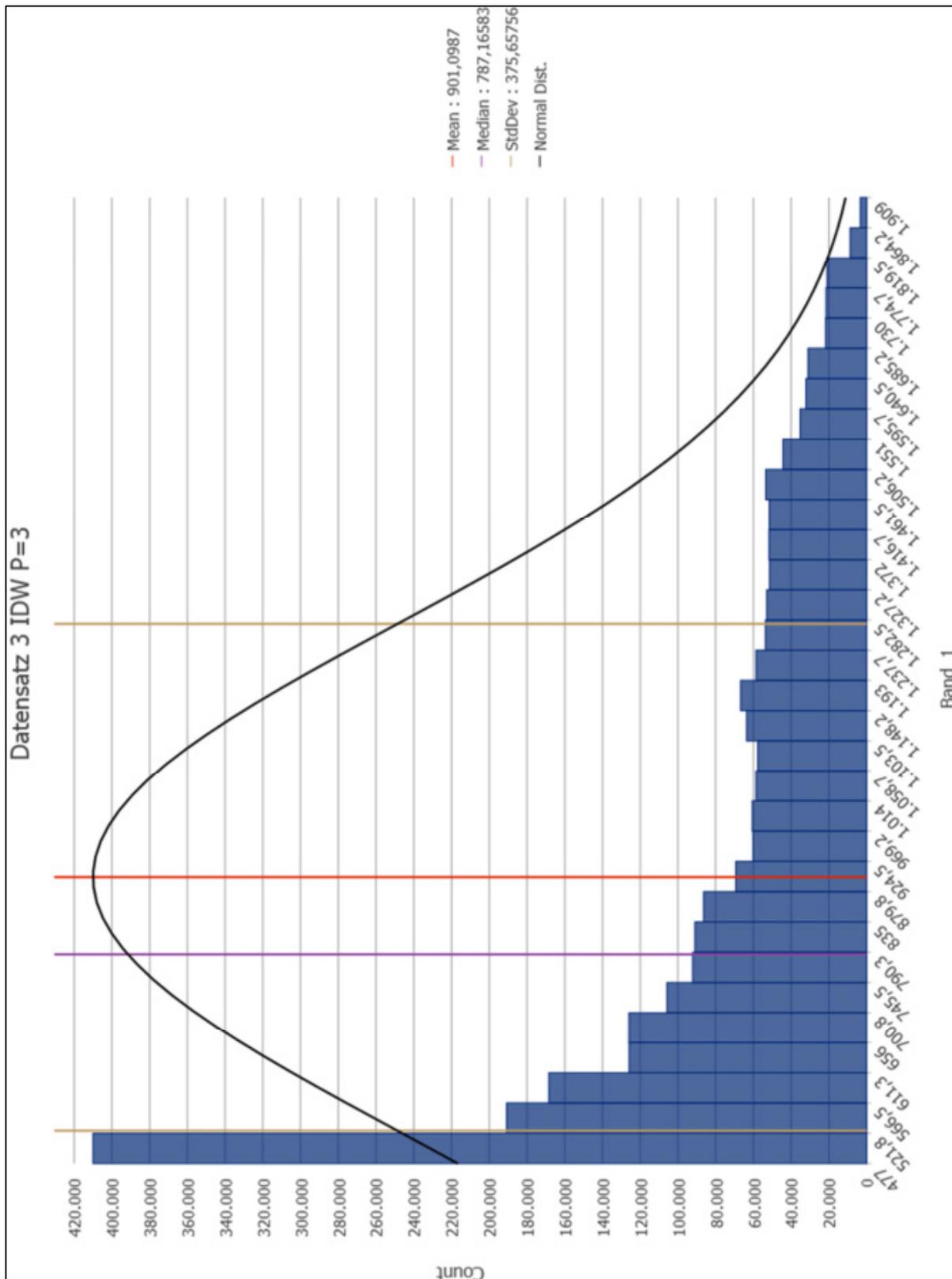


Abbildung 65: Werteverteilung für „IDW“ („Power“ = 3) – Basisdatensatz 3

- „Ordinary Kriging“-Ergebnis mit Semivariogramm = „Exponential“ (Datensatz 3)

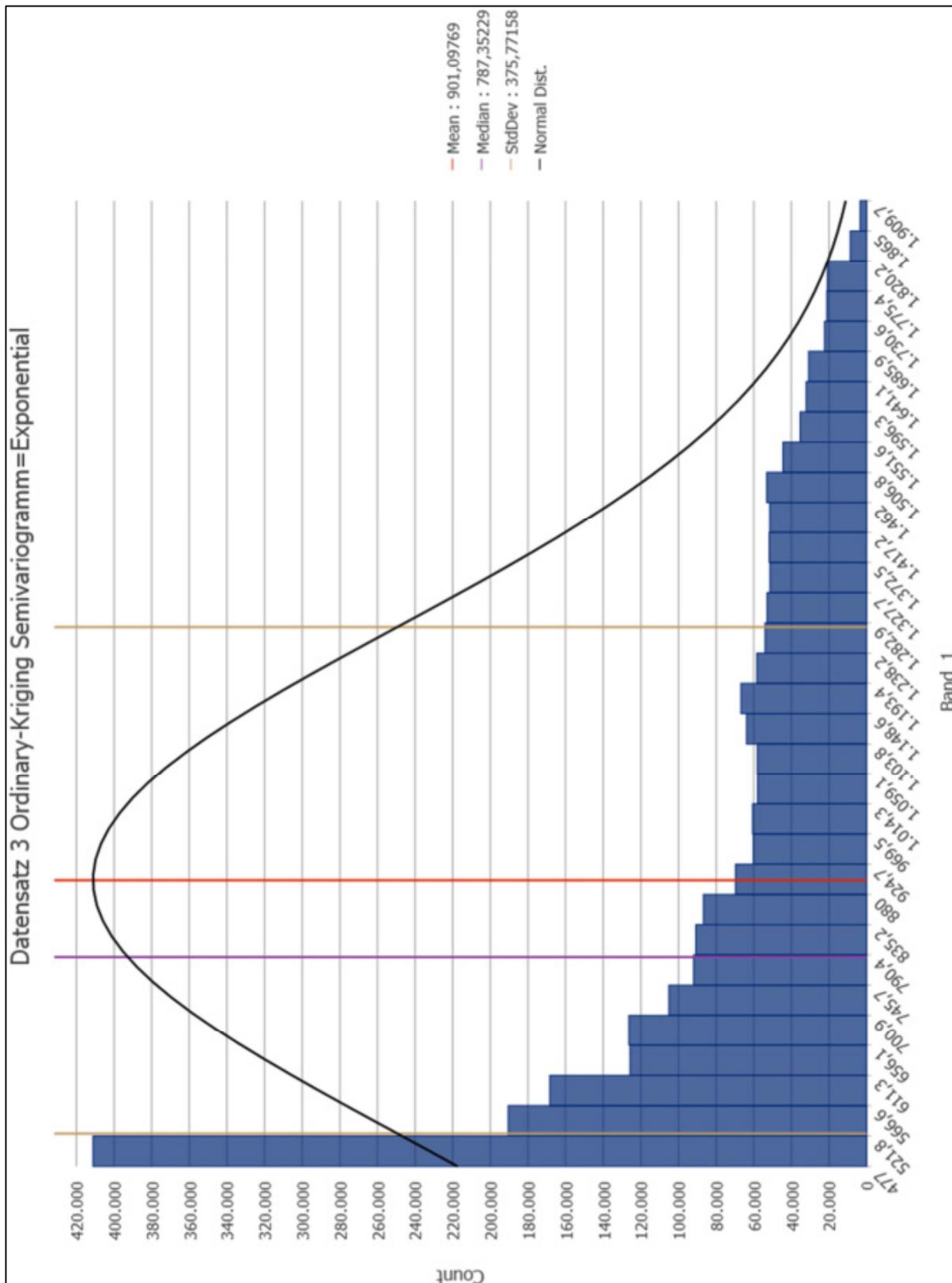


Abbildung 66: Werteverteilung für „Ordinary-Kriging“ („Expon.“) – Basisdatensatz 3

- „Spline“-Ergebnis mit „Weight“-Parameter = 0,5 (Datensatz 3)

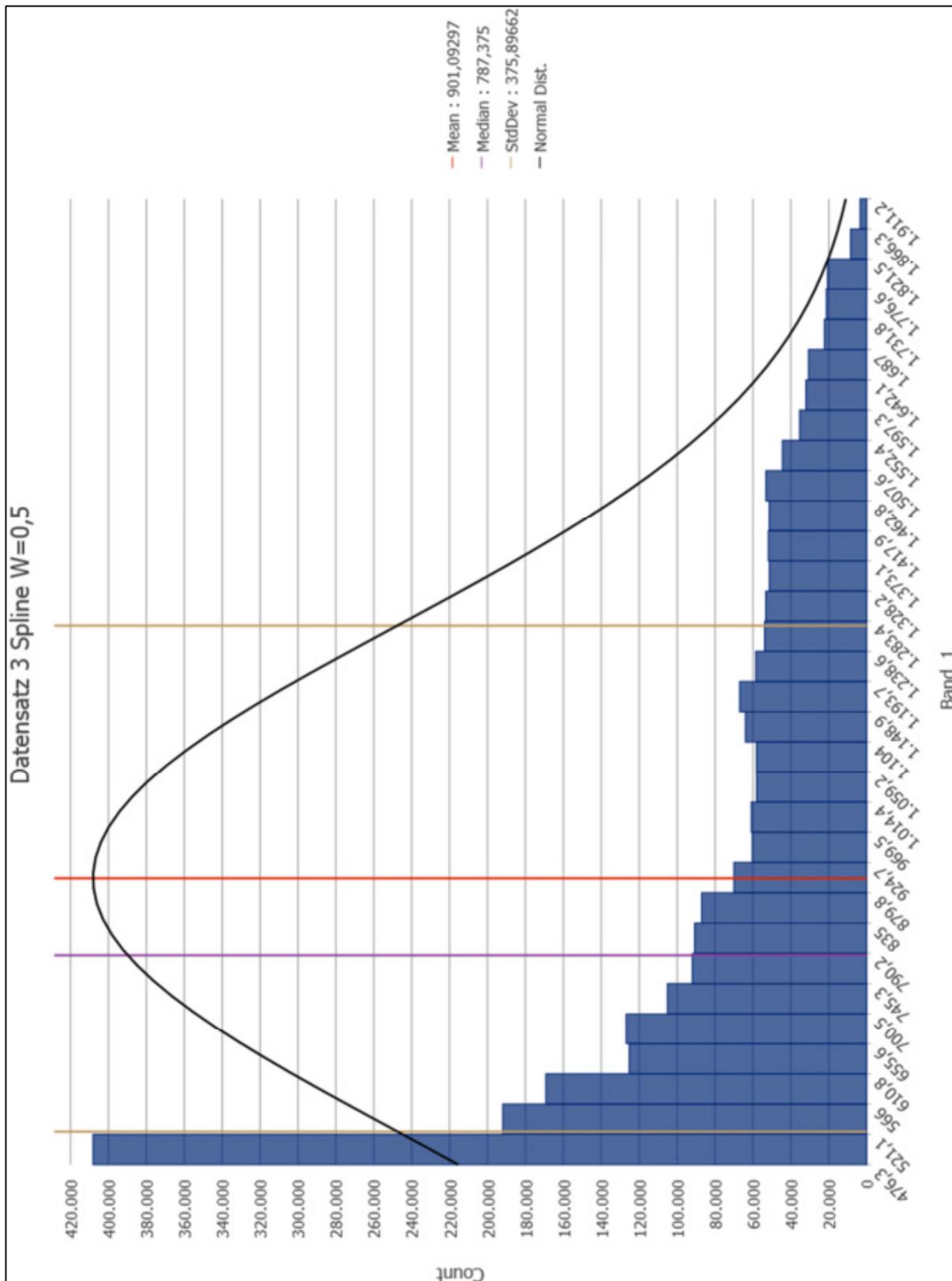


Abbildung 67: Werteverteilung für „Spline“ („Weight“ = 0,5) – Basisdatensatz 3

## Anhang C – Referenzpunkte – berechnete Interpolationswerte

- „IDW“-Zellenwerte – Basisdatensatz 1

POINT	IDW_P25	IDW_P50	IDW_P75	IDW_P100	IDW_P125	IDW_P150	IDW_P175	IDW_P200	IDW_P225	IDW_P250	IDW_P275	IDW_P300
0	514,82	514,81	514,81	514,81	514,81	514,81	514,81	514,81	514,81	514,81	514,81	514,81
1	514,15	514,15	514,14	514,14	514,14	514,14	514,14	514,14	514,14	514,14	514,14	514,14
2	509,87	509,87	509,87	509,87	509,87	509,87	509,87	509,87	509,87	509,87	509,87	509,87
3	509,29	509,28	509,28	509,28	509,28	509,28	509,28	509,28	509,28	509,28	509,27	509,27
4	507,64	507,64	507,64	507,64	507,64	507,64	507,64	507,63	507,63	507,63	507,63	507,63
5	508,21	508,21	508,21	508,21	508,20	508,20	508,20	508,20	508,19	508,19	508,19	508,19
6	511,58	511,59	511,59	511,60	511,60	511,61	511,61	511,62	511,62	511,63	511,63	511,63
7	512,70	512,70	512,70	512,70	512,70	512,70	512,70	512,70	512,71	512,71	512,71	512,71
8	505,82	505,82	505,82	505,82	505,82	505,82	505,82	505,82	505,82	505,82	505,82	505,82
9	509,33	509,33	509,33	509,34	509,34	509,34	509,34	509,34	509,34	509,34	509,34	509,34
10	501,42	501,42	501,42	501,42	501,42	501,42	501,42	501,42	501,42	501,42	501,42	501,42
11	499,29	499,29	499,29	499,28	499,28	499,27	499,27	499,27	499,26	499,26	499,26	499,26
12	498,27	498,27	498,27	498,27	498,27	498,27	498,27	498,27	498,27	498,27	498,27	498,27
13	497,40	497,39	497,37	497,36	497,35	497,34	497,33	497,32	497,31	497,30	497,29	497,29
14	484,97	484,97	484,97	484,96	484,96	484,96	484,96	484,96	484,96	484,96	484,95	484,95
15	484,69	484,69	484,69	484,69	484,68	484,68	484,68	484,68	484,68	484,68	484,68	484,67
16	495,46	495,46	495,46	495,46	495,46	495,46	495,46	495,46	495,46	495,46	495,46	495,46
17	495,42	495,42	495,42	495,42	495,42	495,42	495,42	495,42	495,42	495,42	495,42	495,42
18	484,67	484,67	484,67	484,67	484,67	484,67	484,67	484,67	484,66	484,66	484,66	484,66
19	493,00	493,00	493,00	493,00	492,99	492,99	492,99	492,99	492,99	492,99	492,99	492,99
20	484,08	484,08	484,07	484,07	484,06	484,06	484,06	484,05	484,05	484,05	484,05	484,04
21	517,22	517,22	517,22	517,22	517,21	517,21	517,21	517,21	517,21	517,21	517,21	517,21
22	494,72	494,72	494,72	494,72	494,72	494,72	494,72	494,72	494,72	494,72	494,72	494,72
Daten 1	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW
Daten 1	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW

Tabelle 15: „IDW“-Zellenwerte – Basisdatensatz 1

- „Kriging“-Zellenwerte – Basisdatensatz 1

POINT	SPH	EXP	GAU	LiDr
0	514,79	514,79	514,82	514,82
1	514,14	514,14	514,15	514,15
2	509,87	509,87	509,87	509,87
3	509,26	509,26	509,29	509,29
4	507,61	507,61	507,64	507,64
5	508,16	508,16	508,21	508,21
6	511,69	511,69	511,58	511,58
7	512,71	512,71	512,70	512,70
8	505,82	505,82	505,82	505,82
9	509,35	509,35	509,33	509,33
10	501,43	501,43	501,42	501,42
11	499,21	499,21	499,30	499,30
12	498,27	498,27	498,27	498,27
13	497,14	497,14	497,41	497,41
14	484,93	484,93	484,97	484,97
15	484,65	484,65	484,69	484,69
16	495,47	495,47	495,46	495,46
17	495,41	495,41	495,42	495,42
18	484,65	484,65	484,67	484,67
19	492,98	492,98	493,00	493,00
20	484,00	484,00	484,08	484,08
21	517,20	517,20	517,22	517,22
22	494,72	494,72	494,72	494,72
	Daten 1	Daten 1	Daten 1	Daten 1
	Kriging	Kriging	Kriging	Kriging

Tabelle 16: „Kriging“-Zellenwerte – Basisdatensatz 1

• „Spline“-Zellenwerte – Basisdatensatz 1

POINT	SPL_W5	SPL_W10	SPL_W15	SPL_W20	SPL_W25	SPL_W30	SPL_W35	SPL_W40	SPL_W45	SPL_W50
0	514,78	514,78	514,78	514,78	514,78	514,78	514,78	514,78	514,78	514,78
1	514,14	514,14	514,14	514,14	514,14	514,14	514,14	514,14	514,14	514,14
2	509,88	509,88	509,88	509,88	509,88	509,88	509,88	509,88	509,88	509,88
3	509,27	509,27	509,27	509,27	509,27	509,27	509,27	509,27	509,27	509,27
4	507,60	507,60	507,60	507,60	507,60	507,60	507,60	507,60	507,60	507,60
5	508,03	508,02	508,02	508,02	508,02	508,02	508,02	508,01	508,01	508,01
6	511,71	511,71	511,71	511,71	511,71	511,71	511,71	511,71	511,71	511,71
7	512,71	512,71	512,71	512,71	512,71	512,71	512,71	512,71	512,71	512,71
8	505,82	505,82	505,82	505,82	505,82	505,82	505,82	505,82	505,82	505,82
9	509,22	509,21	509,21	509,20	509,20	509,20	509,20	509,20	509,20	509,20
10	501,43	501,42	501,42	501,42	501,42	501,42	501,42	501,42	501,42	501,42
11	499,21	499,21	499,21	499,21	499,21	499,21	499,21	499,21	499,21	499,21
12	498,27	498,27	498,27	498,27	498,27	498,27	498,27	498,27	498,27	498,27
13	497,20	497,20	497,20	497,20	497,20	497,20	497,20	497,20	497,20	497,20
14	484,93	484,93	484,93	484,93	484,93	484,93	484,93	484,93	484,93	484,93
15	484,63	484,63	484,63	484,63	484,63	484,63	484,63	484,63	484,63	484,63
16	495,47	495,47	495,47	495,47	495,47	495,47	495,47	495,47	495,47	495,47
17	495,41	495,41	495,41	495,41	495,41	495,41	495,41	495,41	495,41	495,41
18	484,64	484,64	484,64	484,64	484,64	484,64	484,64	484,64	484,64	484,64
19	492,98	492,98	492,98	492,98	492,98	492,98	492,98	492,98	492,98	492,98
20	483,96	483,95	483,95	483,95	483,95	483,95	483,95	483,95	483,95	483,95
21	517,18	517,18	517,18	517,18	517,18	517,18	517,18	517,18	517,18	517,18
22	494,72	494,72	494,72	494,72	494,72	494,72	494,72	494,72	494,72	494,72
Daten 1										
Spline										

Tabelle 17: „Spline“-Zellenwerte – Basisdatensatz 1

• „IDW“-Zellenwerte – Basisdatensatz 2

POINT	IDW_P25	IDW_P50	IDW_P75	IDW_P100	IDW_P125	IDW_P150	IDW_P175	IDW_P200	IDW_P225	IDW_P250	IDW_P275	IDW_P300
0	514,85	514,84	514,84	514,84	514,84	514,83	514,83	514,83	514,83	514,83	514,83	514,83
1	512,99	513,02	513,04	513,07	513,09	513,12	513,14	513,16	513,18	513,20	513,22	513,24
2	509,92	509,92	509,91	509,91	509,91	509,91	509,90	509,90	509,90	509,90	509,90	509,89
3	510,05	510,01	509,98	509,94	509,91	509,88	509,84	509,82	509,79	509,76	509,74	509,72
4	508,08	508,05	508,02	507,99	507,96	507,93	507,91	507,89	507,87	507,85	507,83	507,81
5	508,74	508,73	508,73	508,72	508,71	508,71	508,70	508,70	508,69	508,68	508,68	508,67
6	510,16	510,23	510,29	510,35	510,41	510,46	510,52	510,56	510,61	510,65	510,69	510,73
7	512,11	512,16	512,20	512,25	512,29	512,33	512,37	512,41	512,44	512,48	512,51	512,53
8	505,75	505,76	505,76	505,76	505,76	505,76	505,77	505,77	505,77	505,77	505,77	505,77
9	506,36	506,37	506,38	506,40	506,42	506,44	506,46	506,48	506,51	506,53	506,56	506,59
10	501,55	501,55	501,54	501,54	501,53	501,53	501,52	501,52	501,51	501,51	501,50	501,50
11	499,05	499,04	499,03	499,02	499,01	499,01	499,01	499,01	499,01	499,01	499,02	499,02
12	498,25	498,26	498,26	498,26	498,26	498,26	498,26	498,26	498,26	498,26	498,26	498,26
13	499,59	499,58	499,58	499,57	499,56	499,55	499,54	499,53	499,52	499,51	499,50	499,49
14	486,85	486,79	486,74	486,68	486,63	486,57	486,52	486,47	486,43	486,38	486,34	486,30
15	486,15	486,10	486,05	486,01	485,96	485,92	485,88	485,84	485,80	485,76	485,73	485,70
16	495,46	495,46	495,46	495,46	495,46	495,46	495,46	495,46	495,46	495,46	495,46	495,46
17	495,47	495,47	495,47	495,47	495,46	495,46	495,46	495,46	495,46	495,46	495,45	495,45
18	486,55	486,51	486,46	486,42	486,38	486,34	486,30	486,26	486,22	486,19	486,16	486,13
19	493,28	493,27	493,26	493,25	493,24	493,23	493,22	493,21	493,20	493,19	493,18	493,17
20	486,41	486,32	486,24	486,15	486,07	485,99	485,92	485,85	485,78	485,72	485,66	485,61
21	514,17	514,30	514,43	514,55	514,66	514,77	514,86	514,96	515,04	515,12	515,19	515,25
22	494,70	494,70	494,70	494,70	494,70	494,70	494,70	494,71	494,71	494,71	494,71	494,71
Daten 2	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW
Daten 2	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW

Tabelle 18: „IDW“-Zellenwerte – Basisdatensatz 2

- „Kriging“-Zellenwerte – Basisdatensatz 2

POINT	SPH	EXP	GAU	LiDr
0	514,89	514,89	514,73	514,71
1	513,56	513,56	512,96	512,96
2	509,86	509,86	509,92	509,92
3	509,30	509,30	510,08	510,08
4	507,48	507,48	508,11	508,11
5	508,59	508,59	508,75	508,75
6	511,44	511,44	510,10	510,10
7	513,07	513,07	512,06	512,06
8	505,80	505,80	505,75	505,75
9	506,85	506,84	506,35	506,35
10	501,43	501,43	501,56	501,56
11	498,96	498,97	499,06	499,07
12	498,28	498,28	498,25	498,25
13	499,38	499,38	499,59	499,59
14	485,61	485,61	486,91	486,91
15	485,14	485,14	486,20	486,20
16	495,46	495,46	495,45	495,45
17	495,43	495,43	495,47	495,47
18	485,60	485,60	486,59	486,59
19	493,04	493,04	493,30	493,30
20	484,60	484,61	486,49	486,50
21	516,63	516,62	514,05	514,04
22	494,71	494,71	494,70	494,70
	Daten 2	Daten 2	Daten 2	Daten 2
	Kriging	Kriging	Kriging	Kriging

Tabelle 19: „Kriging“-Zellenwerte – Basisdatensatz 2

• „Spline“-Zellenwerte – Basisdatensatz 2

POINT	SPL_W5	SPL_W10	SPL_W15	SPL_W20	SPL_W25	SPL_W30	SPL_W35	SPL_W40	SPL_W45	SPL_W50
0	514,94	514,94	514,94	514,95	514,95	514,95	514,95	514,95	514,95	514,95
1	514,09	514,10	514,11	514,12	514,12	514,12	514,12	514,12	514,12	514,12
2	509,83	509,82	509,82	509,82	509,82	509,82	509,82	509,82	509,82	509,82
3	508,88	508,88	508,88	508,88	508,88	508,88	508,88	508,88	508,88	508,88
4	507,51	507,51	507,51	507,51	507,51	507,51	507,51	507,51	507,52	507,52
5	508,44	508,44	508,44	508,44	508,44	508,44	508,44	508,44	508,44	508,44
6	511,84	511,86	511,87	511,87	511,87	511,87	511,88	511,88	511,88	511,88
7	513,48	513,50	513,51	513,52	513,52	513,53	513,53	513,53	513,53	513,53
8	505,83	505,83	505,83	505,83	505,83	505,83	505,83	505,83	505,83	505,83
9	508,52	508,56	508,58	508,59	508,60	508,60	508,60	508,61	508,61	508,61
10	501,18	501,16	501,15	501,14	501,14	501,13	501,13	501,13	501,13	501,13
11	500,16	500,21	500,23	500,24	500,25	500,25	500,25	500,26	500,26	500,26
12	498,25	498,25	498,25	498,25	498,25	498,25	498,25	498,25	498,25	498,25
13	498,40	498,38	498,37	498,37	498,37	498,36	498,36	498,36	498,36	498,36
14	484,92	484,90	484,89	484,89	484,88	484,88	484,88	484,88	484,88	484,88
15	484,88	484,89	484,90	484,90	484,90	484,90	484,90	484,90	484,90	484,89
16	495,47	495,47	495,47	495,47	495,47	495,47	495,47	495,47	495,47	495,47
17	495,40	495,40	495,40	495,40	495,40	495,40	495,40	495,39	495,39	495,39
18	484,77	484,73	484,72	484,71	484,71	484,70	484,70	484,70	484,70	484,70
19	492,93	492,93	492,93	492,93	492,93	492,93	492,93	492,93	492,93	492,93
20	483,74	483,74	483,75	483,75	483,75	483,75	483,76	483,76	483,76	483,76
21	516,44	516,44	516,44	516,44	516,44	516,44	516,44	516,44	516,44	516,43
22	494,72	494,72	494,72	494,72	494,72	494,72	494,72	494,72	494,72	494,72
Daten 2										
Spline										

Tabelle 20: „Spline“-Zellenwerte – Basisdatensatz 2

• „IDW“-Zellenwerte – Basisdatensatz 3

POINT	IDW_P25	IDW_P50	IDW_P75	IDW_P100	IDW_P125	IDW_P150	IDW_P175	IDW_P200	IDW_P225	IDW_P250	IDW_P275	IDW_P300
0	1762,27	1761,94	1761,61	1761,27	1760,94	1760,60	1760,28	1759,96	1759,66	1759,37	1759,10	1758,84
1	1350,63	1350,01	1349,40	1348,80	1348,20	1347,61	1347,04	1346,47	1345,92	1345,38	1344,85	1344,33
2	1049,00	1048,81	1048,62	1048,41	1048,20	1047,97	1047,75	1047,51	1047,28	1047,04	1046,80	1046,57
3	1423,59	1423,28	1422,99	1422,72	1422,49	1422,27	1422,09	1421,93	1421,80	1421,69	1421,60	1421,54
4	961,77	961,80	961,83	961,86	961,89	961,92	961,95	961,98	962,01	962,04	962,08	962,11
5	1730,96	1731,00	1731,05	1731,10	1731,14	1731,18	1731,22	1731,25	1731,27	1731,29	1731,30	1731,30
6	1366,31	1366,87	1367,43	1367,99	1368,54	1369,08	1369,61	1370,14	1370,64	1371,13	1371,61	1372,07
7	1245,14	1244,44	1243,73	1243,01	1242,28	1241,57	1240,85	1240,15	1239,45	1238,77	1238,10	1237,45
8	1356,14	1356,10	1356,05	1355,99	1355,92	1355,84	1355,74	1355,63	1355,51	1355,38	1355,23	1355,07
9	636,90	636,63	636,35	636,07	635,78	635,48	635,18	634,88	634,57	634,27	633,96	633,66
10	518,89	518,78	518,67	518,56	518,45	518,34	518,24	518,13	518,03	517,93	517,83	517,74
11	499,00	499,00	499,00	499,00	499,00	499,00	499,00	499,00	499,00	499,00	499,00	499,00
12	821,80	821,45	821,12	820,81	820,53	820,27	820,05	819,85	819,69	819,55	819,45	819,37
13	501,30	501,26	501,23	501,20	501,17	501,14	501,11	501,09	501,07	501,05	501,03	501,01
14	713,96	713,99	714,02	714,03	714,04	714,04	714,03	714,01	713,99	713,96	713,93	713,90
15	724,57	724,71	724,84	724,96	725,06	725,16	725,24	725,32	725,39	725,46	725,52	725,59
16	651,99	651,97	651,94	651,91	651,88	651,85	651,82	651,79	651,76	651,74	651,73	651,72
17	553,00	553,01	553,02	553,02	553,03	553,04	553,05	553,06	553,07	553,08	553,09	553,10
18	509,85	509,86	509,88	509,90	509,92	509,95	509,98	510,01	510,04	510,08	510,11	510,15
19	738,57	738,55	738,54	738,53	738,53	738,52	738,51	738,51	738,50	738,49	738,48	738,46
20	501,02	500,96	500,90	500,83	500,78	500,72	500,67	500,62	500,57	500,52	500,48	500,44
21	1145,85	1145,38	1144,93	1144,50	1144,09	1143,70	1143,34	1143,00	1142,69	1142,40	1142,14	1141,90
22	539,63	539,68	539,72	539,77	539,81	539,85	539,90	539,94	539,98	540,01	540,05	540,09
Daten 3	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW
Daten 3	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW	IDW

Tabelle 21: „IDW“-Zellenwerte – Basisdatensatz 3

- „Kriging“-Zellenwerte – Basisdatensatz 3

POINT	SPH	EXP	GAU	LiDr
0	1762,24	1752,36	1762,58	1763,66
1	1349,58	1343,30	1351,25	1348,43
2	1048,71	1045,30	1049,17	1048,40
3	1423,49	1416,77	1423,92	1424,67
4	961,80	962,34	961,75	961,79
5	1730,78	1734,23	1730,92	1729,87
6	1367,04	1375,98	1365,75	1367,23
7	1244,27	1232,17	1245,83	1244,15
8	1355,67	1358,73	1356,17	1353,98
9	636,34	633,77	637,17	635,38
10	518,72	517,26	519,00	518,60
11	499,00	499,00	499,00	499,00
12	821,81	813,20	822,17	823,64
13	501,25	500,78	501,33	501,24
14	714,02	713,60	713,92	714,06
15	724,92	724,16	724,42	725,56
16	652,25	648,82	652,00	653,37
17	553,04	553,06	553,00	553,14
18	509,88	510,08	509,83	509,96
19	738,44	739,75	738,58	738,02
20	500,96	499,83	501,08	501,05
21	1145,52	1136,47	1146,33	1146,51
22	539,74	539,75	539,58	539,95
	Daten 3	Daten 3	Daten 3	Daten 3
	Kriging	Kriging	Kriging	Kriging

Tabelle 22: „Kriging“-Zellenwerte – Basisdatensatz 3

• „Spline“-Zellenwerte – Basisdatensatz 3

POINT	SPL_W5	SPL_W10	SPL_W15	SPL_W20	SPL_W25	SPL_W30	SPL_W35	SPL_W40	SPL_W45	SPL_W50
0	1737,92	1737,34	1737,11	1736,98	1736,90	1736,84	1736,80	1736,77	1736,75	1736,73
1	1344,24	1344,26	1344,27	1344,29	1344,29	1344,30	1344,30	1344,31	1344,31	1344,32
2	1037,84	1037,53	1037,40	1037,33	1037,28	1037,24	1037,22	1037,20	1037,18	1037,17
3	1417,15	1417,36	1417,46	1417,51	1417,55	1417,58	1417,60	1417,62	1417,64	1417,65
4	963,32	963,28	963,26	963,25	963,24	963,24	963,23	963,23	963,23	963,23
5	1737,96	1738,09	1738,14	1738,18	1738,20	1738,21	1738,22	1738,23	1738,23	1738,24
6	1385,17	1385,60	1385,77	1385,87	1385,93	1385,98	1386,01	1386,03	1386,05	1386,07
7	1220,27	1219,76	1219,56	1219,44	1219,37	1219,32	1219,29	1219,26	1219,24	1219,22
8	1361,41	1361,40	1361,38	1361,37	1361,36	1361,35	1361,34	1361,34	1361,33	1361,32
9	631,43	631,32	631,27	631,24	631,22	631,21	631,20	631,19	631,18	631,18
10	517,43	517,44	517,45	517,46	517,46	517,47	517,47	517,47	517,48	517,48
11	499,10	499,09	499,09	499,09	499,09	499,09	499,09	499,09	499,09	499,09
12	813,68	813,87	813,96	814,01	814,05	814,08	814,10	814,12	814,13	814,14
13	501,21	501,22	501,23	501,23	501,23	501,23	501,23	501,23	501,23	501,23
14	709,63	709,38	709,26	709,19	709,14	709,11	709,08	709,06	709,05	709,03
15	716,12	715,88	715,77	715,72	715,68	715,65	715,63	715,61	715,60	715,59
16	642,99	642,82	642,75	642,72	642,69	642,68	642,67	642,66	642,65	642,65
17	559,92	554,02	554,07	554,09	554,11	554,12	554,13	554,14	554,14	554,15
18	510,58	510,62	510,64	510,66	510,67	510,68	510,68	510,69	510,69	510,70
19	743,87	744,01	744,07	744,10	744,13	744,15	744,16	744,17	744,18	744,19
20	499,55	499,57	499,58	499,59	499,59	499,59	499,59	499,60	499,60	499,60
21	1133,10	1133,20	1133,26	1133,30	1133,33	1133,35	1133,36	1133,38	1133,39	1133,40
22	538,43	538,36	538,33	538,31	538,30	538,29	538,29	538,28	538,28	538,28
Daten 3										
Spline										

Tabelle 23: „Spline“-Zellenwerte – Basisdatensatz 3

## Anhang D – „Nearest Neighbor“-Analysen der drei Basisdatensätze

- „Nearest Neighbor“-Analyse („NNA“) – Basisdatensatz 1

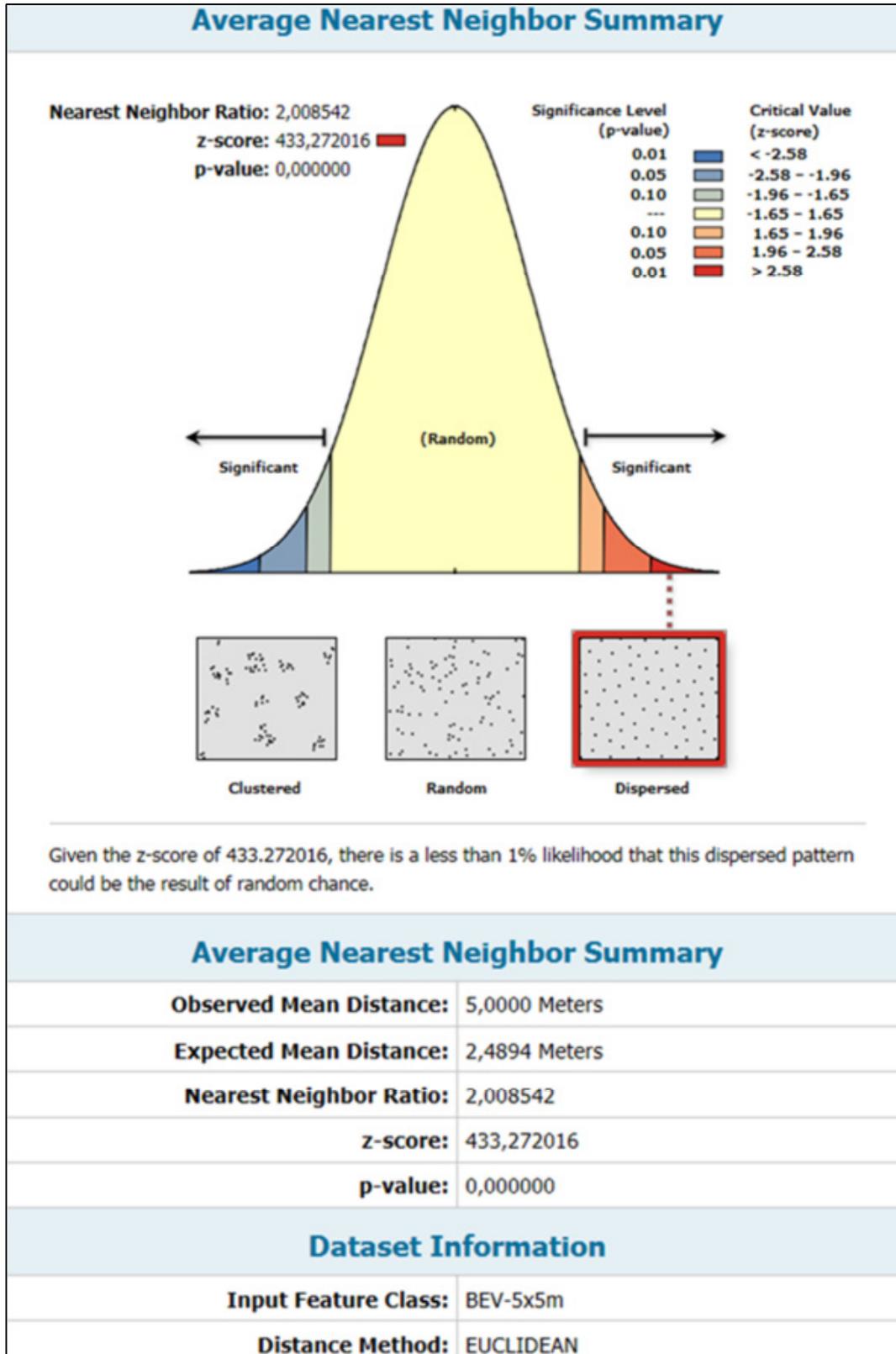


Abbildung 68: „Nearest Neighbor“-Analyse – Basisdatensatz 1

- „Nearest Neighbor“-Analyse („NNA“) – Basisdatensatz 2

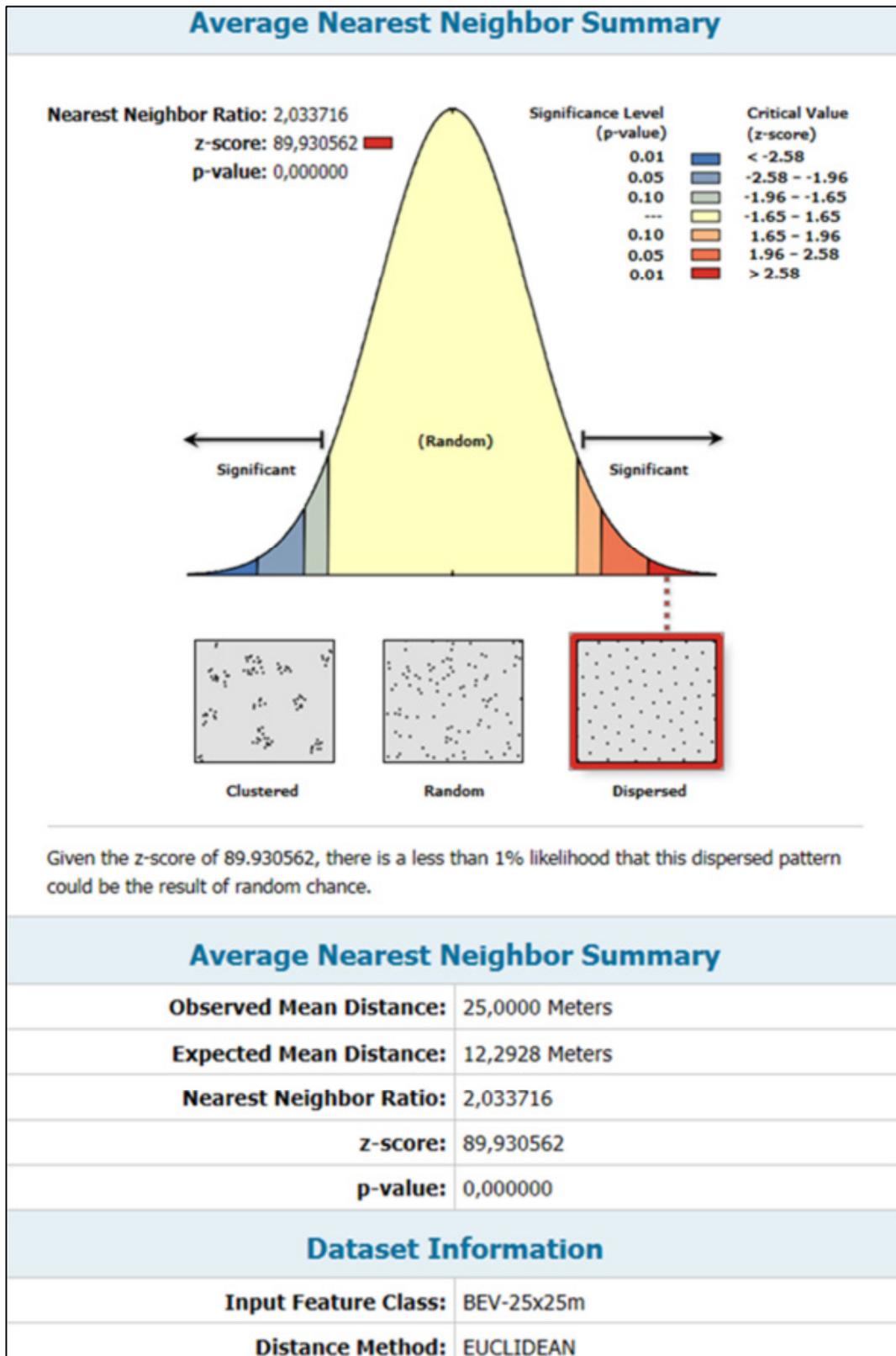


Abbildung 69: „Nearest Neighbor“-Analyse – Basisdatensatz 2

• „Nearest Neighbor“-Analyse („NNA“) – Basisdatensatz 3

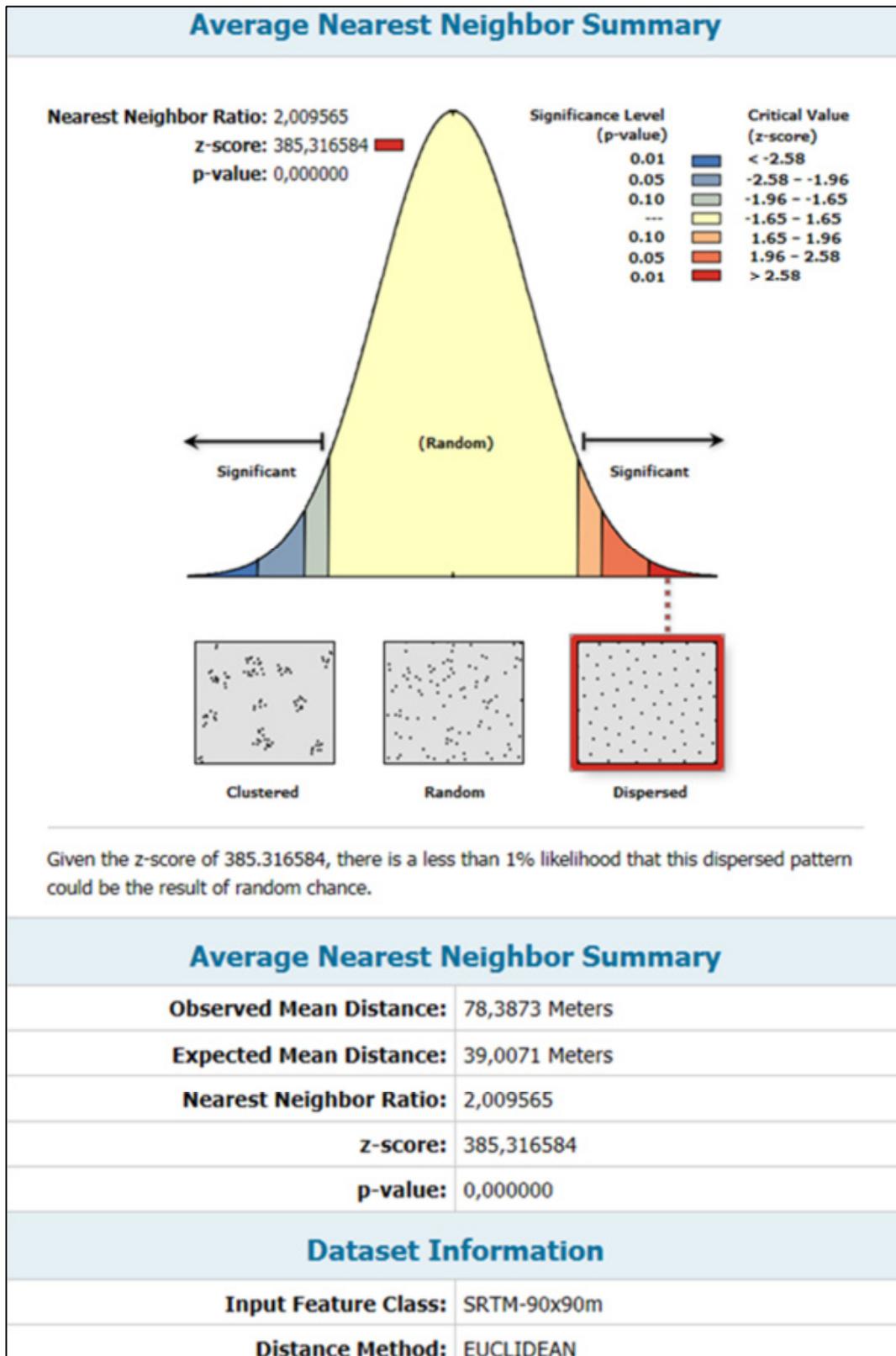


Abbildung 70: „Nearest Neighbor“-Analyse – Basisdatensatz 3