

Hochschule für Angewandte Wissenschaften Hamburg

Fakultät Life Sciences

**Treibverhalten von vermissten Personen am Beispiel
des Hamburger Hafens**

Bachelorarbeit

im Studiengang Rettungsingenieurwesen

Manuel Wiesenmaier



Gutachter:	Prof. B. Tolg	(HAW Hamburg)
Gutachter:	M.Sc. J. Steinebrei	(Feuerwehr Hamburg)

Hamburg

den 21.06.2022

Inhaltsverzeichnis

Inhaltsverzeichnis.....	2
Abbildungsverzeichnis	4
Tabellenverzeichnis.....	5
Abkürzungsverzeichnis.....	5
1 Einleitung und Ziel.....	6
1.1 Einleitung	6
1.2 Ziel und Aufbau der Arbeit.....	7
2 Grundlagen.....	9
2.1.1 Die Elbe im Raum Hamburg	9
2.2 Physikalische Grundlagen	10
2.2.1 Strömungswiderstand	10
2.2.2 Aufschwimmen	12
2.3 Konvertierung der UTM-Koordinaten in geographische Kugelkoordinaten.....	13
3 Material und Methoden.....	15
3.1 Material.....	15
3.2 Methode	17
3.2.1 Recherche	17
3.2.2 Bestimmung der Konstanten A , c_w , M und ρ	17
3.2.3 Numerische Diskretisierung.....	18
3.2.4 Trajektorien Berechnung	18
3.2.5 Streuung des Startpunktes.....	20
3.2.6 Umsetzung in C#	20
4 Ergebnisse	38
4.1 Der Menschliche Körper auf und unter Wasser	38
4.1.1 Schwimmfähigkeit des menschlichen Körpers.....	38
4.1.2 Einfluss der Bodenreibung auf die Bewegung	39
4.1.3 Einfluss der Wassertemperatur auf den menschlichen Körper	39
4.1.4 Absinken.....	40
4.1.5 Aufschwimmen	42
4.2 Auswertung der Simulationen	43
5 Diskussion	58
5.1 Verweildauer eines menschlichen Körpers auf und unter Wasser.....	58
5.1.1 Auswertung der Modellierung	60
6 Fazit und Ausblick	61

7	Anhang	63
8	Literaturverzeichnis	64

Abbildungsverzeichnis

ABBILDUNG 1 SEITENSICHTSONAR	6
ABBILDUNG 2 BETRAG DER STRÖMUNGSGESCHWINDIGKEIT ÜBER DIE ZEIT (6)	10
ABBILDUNG 3 REIBUNGS- UND DRUCKWIDERSTAND (7)	10
ABBILDUNG 4 BEREICH FÜR DEN STRÖMUNGSDATEN VORLIEGEN	16
ABBILDUNG 5 BERECHNUNG DER GEOGRAPHISCHEN POSITION	19
ABBILDUNG 6 VERTEILUNG DER STARTPUNKTE FÜR DIE SIMULATION.....	20
ABBILDUNG 7 PROGRAMMCODE	22
ABBILDUNG 8 PROGRAMMCODE	23
ABBILDUNG 9 PROGRAMMCODE	23
ABBILDUNG 10 PROGRAMMCODE	24
ABBILDUNG 11 PROGRAMMCODE	25
ABBILDUNG 12 PROGRAMMCODE	25
ABBILDUNG 13 PROGRAMMCODE	25
ABBILDUNG 14 PROGRAMMCODE	26
ABBILDUNG 15 PROGRAMMCODE	27
ABBILDUNG 17 PROGRAMMCODE	28
ABBILDUNG 16 PROGRAMMCODE	28
ABBILDUNG 19 PROGRAMMCODE	29
ABBILDUNG 18 PROGRAMMCODE	29
ABBILDUNG 20 PROGRAMMCODE	30
ABBILDUNG 21 PROGRAMMCODE	30
ABBILDUNG 22 PROGRAMMCODE	31
ABBILDUNG 23 PROGRAMMCODE	31
ABBILDUNG 24 PROGRAMMCODE	32
ABBILDUNG 25 PROGRAMMCODE	32
ABBILDUNG 26 PROGRAMMCODE	32
ABBILDUNG 27 PROGRAMMCODE	33
ABBILDUNG 28 PROGRAMMCODE	34
ABBILDUNG 29 IM WEBBROWSER GEÖFFNETE KARTE	34
ABBILDUNG 30 PROGRAMMCODE	35
ABBILDUNG 31 PROGRAMMCODE	36
ABBILDUNG 32 PROGRAMMCODE	37
ABBILDUNG 33: AUF DEN GRUND GESUNKENE PERSON.....	39
ABBILDUNG 34SIMULATION MAHATMA-GANDHI-BRÜCKE.....	47
ABBILDUNG 35 SIMULATION KÖLBRANDBRÜCKE	49
ABBILDUNG 36 SIMULATION VEDDELER KANAL	51
ABBILDUNG 37 SIMULATION ELBPARK ENTENWERDER.....	53
ABBILDUNG 38 SIMULATION FLUGHAFEN HAMBURG FINKENWERDER.....	55

ABBILDUNG 39 SIMULATION ÖVELGÖNNE.....	57
ABBILDUNG 40 VERWEILDAUER IN ABHÄNGIGKEIT DER ZEIT.....	60

Tabellenverzeichnis

TABELLE 1 ZEIT BIS KÖRPERTEMPERATUR AUF 35 °C ABKÜHLT IN ABHÄNGIGKEIT DER WASSERTEMPERATUR.....	40
TABELLE 2: ABSINKZEITEN IN ABHÄNGIGKEIT DER TIEFE FÜR EINE PERSON, MIT EINEM KÖRPERGEWICHT VON 80KG, EINER GRÖÖE VON 1,8M.....	42
TABELLE 3 ZUSAMMENHANG ZWISCHEN WASSERTEMPERATUR UND ADD	43
TABELLE 4 SIMULATION MAHATMA-GANDHI-BRÜCKE DURCHLAUF 1	44
TABELLE 5 SIMULATION MAHATMA-GANDHI-BRÜCKE DURCHLAUF 2	45
TABELLE 6 SIMULATION MAHATMA-GANDHI-BRÜCKE DURCHLAUF 3	46
TABELLE 8 SIMULATION KÖHLBRANDBRÜCKE.....	48
TABELLE 9 SIMULATION VEDDELER KANAL	50
TABELLE 10 SIMULATION ELBPARK ENTENWERDER	52
TABELLE 11 SIMULATION FLUGPLATZ HAMBURG FINKENWERDER	54
TABELLE 12 SIMULATION ÖVELGÖNNE	56

Abkürzungsverzeichnis

ADD <i>accumulated degree day</i>	UTM
C# <i>engl. C-Sharp</i>	Universale Transversale Merkatorprojektion
GRS 80 <i>Geodätischen Referenzsystem 1980</i>	18
HPA <i>Hamburger Port Authority</i>	WMS <i>Web Map Services</i>
OpCIS <i>Operational Current Informations System</i>	

1 Einleitung und Ziel

1.1 Einleitung

Die Suche nach vermissten Personen in Gewässern ist mit einem großen Personenaufwand verbunden, da große Wasserflächen abgesucht werden müssen. Die Suche verläuft sowohl an Land als auch auf bzw. im Wasser, da häufig unklar ist, ob die Person aus eigener Kraft an Land gekommen ist. Durch Befragung der meldenden Person wird versucht, den Standort zu ermitteln, an dem die vermisste Person das letzte Mal gesichtet wurde. Mithilfe von Rettungshubschraubern wird anschließend die Wasseroberfläche des Gewässers durchsucht.

Ist davon auszugehen, dass die Person sich Unterwasser befindet, ist die Suche Unterwasser mithilfe von Rettungstauchern nötig. Diese werden in Hamburg mit einem Seitensichtsonar („side-scan sonar) unterstützt (1). Die Sonargeräte funktionieren ähnlich wie medizinische Ultraschallgeräte und sind entweder fest am Bootsrumpf befestigt oder werden „mobil“ über die Bootswand mit einer Stange in das Wasser getaucht (2). Die Auflösung, der mit dem Sonar dargestellten Bilder, wird mit steigender Frequenz besser, wobei dabei gleichzeitig die Eindringtiefe sinkt. Somit verringert sich die Qualität der Bilder mit steigender Tiefe des Gewässers. Eine ausreichende Auflösung, um eine Person bzw. ein Objekt in der Größe einer Person zu erkennen, ist bis zu einer Tiefe von 15-20 m über dem Grund möglich. Ein mögliches Bild aus einem Seitensichtsonar ist in Abbildung 1 dargestellt.

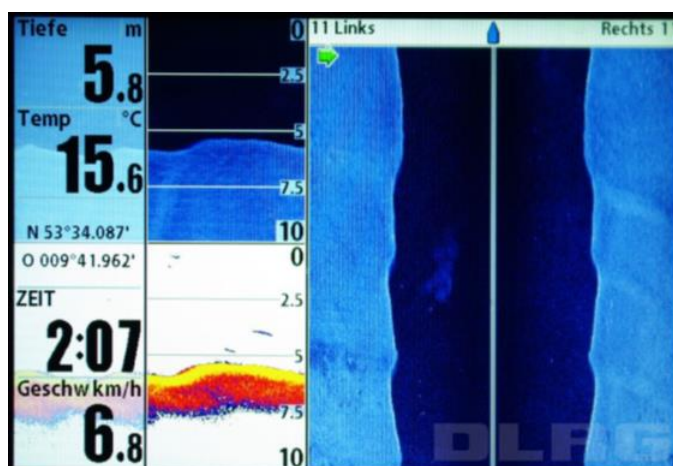


Abbildung 1 Seitensichtsonar

Die Auswertung der Bilder ist schwierig und eine eindeutige Erkennung einer Person ist meist nicht möglich. Deshalb muss die Stelle nachträglich oft von Tauchern untersucht werden (1).

Die Personensuche in fließenden Gewässern wie der Elbe gestaltet sich als besonders schwierig. Grund hierfür ist, dass eine ertrinkende Person aufgrund der Strömung schon nach 10 Minuten mehrere hundert bis tausend Meter abtreiben kann (3). Da die Person nach der letzten Sichtung abgetrieben sein könnte, wird bei der Personensuche in fließenden Gewässern an zwei Stellen gleichzeitig getaucht. Ein Taucheinsatz wird an der vermeintlichen Stelle der letzten Sichtung durchgeführt, während der zweite Taucheinsatz stromabwärts durchgeführt wird. Die Größe des Suchsektors wird anhand der Strömungsgeschwindigkeit abgeschätzt. Da sich aufgrund der Gezeiten die Strömung in der Elbe sowohl ihre Geschwindigkeit als auch die Richtung mehrmals am Tag ändert, sind aktuelle Strömungsdaten zu beachten. Eine Abschätzung des Suchsektors ist bisher nur auf Grundlage von Erfahrungen und ohne Hilfestellungen möglich. Dabei ist eine präzise Abschätzung wichtig, da so der Suchradius verringert und somit wertvolle Zeit gespart werden kann. Ein Modell, das mögliche Standorte berechnet, an denen die Person abgetrieben werden könnte, soll die Suche nach vermissten Personen erheblich erleichtern.

1.2 Ziel und Aufbau der Arbeit

Ziel der Arbeit ist die Entwicklung eines Modells, mit dem das Treibverhalten von vermissten Personen im Hamburger Hafen dargestellt werden kann. Das Modell soll anhand aktueller Strömungsdaten berechnen, wie eine Person durch die Strömung bewegt wird. Nach Auswertung der Berechnung anhand des Modells soll eine Einschätzung über das Treibverhalten abgelesen werden können. Durch Abschätzung des Standorts und Suchradius' kann so die Wahrscheinlichkeit, die zu vermissende Person lebend zu finden, erhöht werden.

Zur Erstellung eines solchen Modells wird in Kapitel 2 zunächst das Schwimmverhalten von menschlichen Körpern und das Verhalten eines menschlichen Körpers in der Strömung erklärt. Des Weiteren werden die Besonderheiten der Elbe im Raum Hamburg erläutert. Es werden die physikalischen Grundlagen des Modells und die Umrechnung von UTM Koordinaten in Kugelkoordinaten dargestellt.

Das Kapitel 3 befasst sich mit dem System, das die nötigen Strömungsdaten für das Modell liefert und illustriert die Methodik der Arbeit. Dabei wird die numerische Diskretisierung und die Umsetzung des Modells in C# (engl. C-Sharp) vorgestellt.

Den Kern der Arbeit bilden Kapitel 4 und 5. Hier werden die Ergebnisse bezüglich des Verhaltens eines menschlichen Körpers im Wasser illustriert und die Ergebnisse der Simulationen anhand des Modells aufgezeigt. Außerdem werden Schlüsse aus den vorgestellten Ergebnissen gezogen.

Im 6. Kapitel werden die gewonnenen Erkenntnisse dieser Arbeit zusammengefasst und ein Fazit gezogen. Des Weiteren werden im Ausblick mehrere Wege, zur weiteren Entwicklung und Verbesserung des Modells, und dessen Möglichkeit zur Implementierung, vorgestellt.

2 Grundlagen

Für die Beurteilung der in dieser Arbeit vorgestellten Modelle werden in den nachfolgenden Abschnitten Grundlagen zum Verständnis der Ergebnisse dargelegt. Neben einem Überblick über das Vorgehen bei der Suche nach vermissten Personen in fließenden Gewässern und die Besonderheiten der Elbe im Raum Hamburg, werden physikalische Grundlagen, sowie die Umrechnung von verschiedenen Koordinatensystemen vorgestellt.

2.1.1 Die Elbe im Raum Hamburg

Im Raum Hamburg hat die Elbe eine Fahrrinntiefe von 16,70 Metern (4). Des Weiteren ist in Hamburg mit kalten Wassertemperaturen zu rechnen. So hat die Elbe an der Bunthauspitze an 221 Tagen im Jahr eine Wassertemperatur von unter 15 °C (5). Diese fällt davon an 97 Tagen im Jahr auf 5°C (5).

Die Strömung der Elbe im Raum Hamburg wird hauptsächlich durch die von See her einschwingende Tidewelle und dem vom Binnenland kommenden Oberwassereinfluss bestimmt (6). Zusätzlich dazu wird die Strömung durch meteorologische Einflüsse und lokale Windverhältnisse beeinflusst. Während der Einfluss der Tidewelle auf Grundlage der Stellung von Erde, Mond und Sonne in weiter Zukunft vorhersehbar ist, können die Einflüsse aus Oberwassereinfluss, Niederschlag und Wind nur für die nahe Zukunft abgeschätzt werden. Aufgrund der unterschiedlichen Einflüsse kommt es im Raum Hamburg zu kurzfristigen und saisonalen Schwankungen der Strömung. Die Dauer einer Tide beträgt dabei 12 Stunden und 25 Minuten. Sie ergibt sich aus der Summe der Flutdauer bzw. der Dauer des auflaufenden Wassers und der Ebbedauer bzw. der Dauer des ablaufenden Wassers. Die Flutdauer ist in jedem Standort unterschiedlich und nimmt stromaufwärts weiter ab. So ist beispielsweise die Flutdauer am Pegel St. Pauli 5 Stunden und 3 Minuten und die Ebbedauer 7 Stunden und 22 Minuten lang.

Aus diesem Grund kann innerhalb eines Tages die Strömung ihre Richtung um 180 Grad ändern. Die Flutstromgeschwindigkeit ist während des zulaufenden Wassers nicht konstant, sondern nimmt anfangs wie in Abbildung 2 dargestellt stetig zu, um dann wieder langsam abzufallen bis ihre Richtung durch Einsetzen des Ebbestroms wechselt. Auch die Ebbestromgeschwindigkeit nimmt anfangs schnell zu, um später bis zum Richtungswechsel abzunehmen, wobei die maximale Ebbestromgeschwindigkeit langsamer ist als die maximale Flutgeschwindigkeit. Da die Strömungsgeschwindigkeit stark von der Flut- und Ebbestromgeschwindigkeit beeinflusst wird, kommt es auch hier zu kurzfristigen hohen Strömungsgeschwindigkeiten.

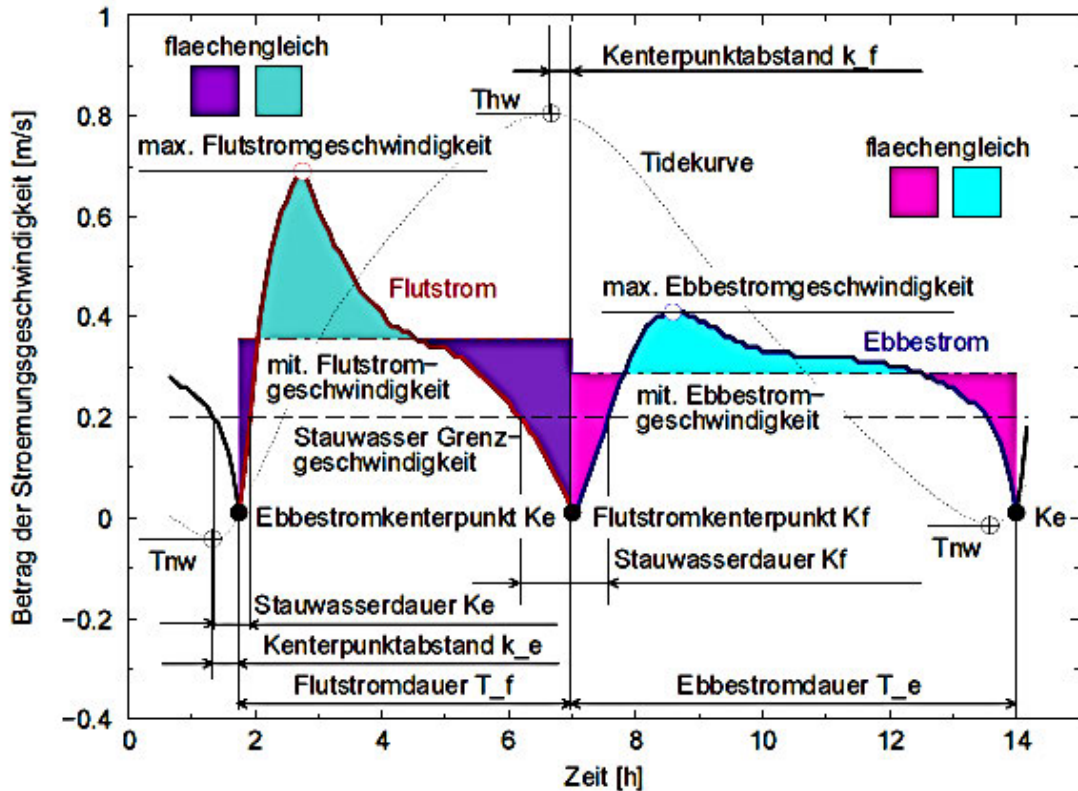


Abbildung 2 Betrag der Strömungsgeschwindigkeit über die Zeit (6)

2.2 Physikalische Grundlagen

2.2.1 Strömungswiderstand

Auf einen sich bewegenden oder ruhenden, von einem Fluid umströmten Körper, werden Kräfte ausgeübt (7). Für stumpfe Körper ist die maßgebende Kraft die Widerstandskraft F_W welche die Richtung der relativen Anströmungsgeschwindigkeit w_∞ hat. Die Widerstandskraft setzt sich aus der Druckspannung F_p und dem Reibungswiderstand F_R zusammen, siehe Abbildung 3.

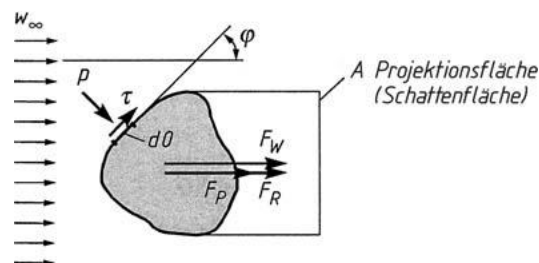


Abbildung 3 Reibungs- und Druckwiderstand (7)

Da eine Vorausberechnung von F_W meist sehr aufwendig ist, wird F_W in Versuchen bestimmt. Dabei wird sich die Reynold'sche Ähnlichkeitstheorie zu Nutze gemacht.

Im Allgemeinen gilt für die Widerstandskraft:

2.2.1.1

$$F_w = c_w \cdot p_d \cdot A$$

Wobei c_w ein dimensionsloser Widerstandswert, p_d der dynamische Druck und A die Projektionsfläche des Körpers auf einer Ebene normal zur Anströmungsgeschwindigkeit ist. Für p_d gilt:

2.2.1.2

$$p_d = \frac{1}{2} \cdot \rho \cdot w_\infty^2$$

ρ ist die Dichte des Fluids. Da es sich bei w_∞ um die relative Strömungsgeschwindigkeit handelt gilt:

2.2.1.3

$$w_\infty = w - v$$

Wobei w die Strömungsgeschwindigkeit und v die Geschwindigkeit des Körpers parallel zur Strömungsrichtung ist.

Setzt man nun 2.2.1.2 und 2.2.1.3 in 2.2.1.1 ein, ergibt sich:

2.2.1.4

$$F_w = c_w \cdot \frac{1}{2} \cdot \rho \cdot |w-v| \cdot (w-v) \cdot A$$

Für eine Kraft F gilt die allgemeine Formel:

2.2.1.5

$$F = M \cdot a$$

Wobei a die Beschleunigung in m/s^2 und M die Masse in kg ist.

Die Beschleunigung kann als erste zeitliche Ableitung der Geschwindigkeit v nach der Zeit dargestellt werden:

2.6

$$a = \frac{dv}{dt}$$

F_w soll gleich F sein. Nach Einsetzen der Gleichung 2.2.1.5 in 2.2.1.4 und der Umformung nach a gilt:

2.7

$$a = \frac{1}{M} \cdot c_w \cdot \frac{1}{2} \cdot \rho \cdot |w-v| \cdot (w-v) \cdot A$$

Es ergibt sich folgende Grundgleichung zu Bestimmung der Geschwindigkeit nach der Zeit:

2.8

$$\frac{dv}{dt} = \frac{1}{M} \cdot c_w \cdot \frac{1}{2} \cdot \rho \cdot |w-v| \cdot (w-v) \cdot A$$

2.2.2 Aufschwimmen

Gemäß des Archimedischen Prinzips sinkt ein Körper dann, wenn dieser mehr wiegt, als die von ihm verdrängte Flüssigkeit (8). In anderen Worten: ein Körper sinkt dann, wenn die Gewichtskraft des Körpers F_k größer ist, als die Gewichtskraft der verdrängten Flüssigkeit F_{fl} . Daraus folgt:

2.2.2.1

$$F_K = V_K \cdot \rho_K \cdot g$$

$$F_{fl} = V_{fl} \cdot \rho_{fl} \cdot g$$

$$V_K \cdot \rho_K \cdot g > V_{fl} \cdot \rho_{fl} \cdot g$$

Wobei V_K das Volumen des Körpers, ρ_K die Dichte des Körpers, g die Erdbeschleunigung mit $9,81\text{m/s}^2$, V_{fl} das Volumen der verdrängten Flüssigkeit und ρ_{fl} die Dichte der Flüssigkeit ist.

Geht man nun davon aus, dass ein Körper sinkt, so entspricht das Volumen der verdrängten Flüssigkeit dem Volumen des Körpers. Es ergibt sich für einen sinkenden Körper:

$$\rho_K > \rho_{fl}$$

2.3 Konvertierung der UTM-Koordinaten in geographische Kugelkoordinaten

Das Universal Transverse Mercator System (UTM) teilt die Erde in 60 Zonen auf (9). Hamburg liegt in der Zone 32. Da es sich beim UTM System um ein rechtwinkliges Koordinatensystem handelt, muss die Erdkrümmung bei Berechnungen innerhalb der Zonen nicht beachtet werden.

Viele Online-Kartendienste nutzen jedoch geographische Kugelkoordinaten für die Beschreibung der Standorte. So nutzt auch der in dieser Arbeit genutzte Online-Kartendienst Geoplaner, bei der Verlinkung der Wegpunkte, Breiten- und Längengradangaben.

Um die Bewegung auf Geoplaner anzeigen zu lassen, müssen die UTM Koordinaten in Längen- und Breitengrade konvertiert werden. Die Konvertierung wird gemäß dem Paper „Map Projektion – A Working Manual“ von John P. Snyder durchgeführt (10). Der Äquator-Radius a wird auf 6.378.137 m und der polar Radius b auf 6356752,3 m gemäß dem Geodätischen Referenzsystem 1980 (GRS 80) gesetzt. Der Maßstabsfaktor k_0 wird auf 0,9996 gesetzt und λ zu Null angenommen. Es wurden die Bezeichnungen nach Snyder genutzt, mit der Ausnahme der Einführung von e_q für e^2 und $x_{-500000}$ anstelle des um 500.000 verringerten x Wertes. Des Weiteren wird der Winkel ϕ_1 im Bogenmaß angegeben. Die Variable x entspricht dem Ostwert der UTM-Koordinaten und die Variable y dem Nordwert der UTM-Koordinaten. Zunächst werden die Hilfsgrößen e, e_q, e_1, e'^2 anhand der zuvor festgelegten Radien a und b berechnet.

2.2.2.1

$$e = \frac{\sqrt{a^2 - b^2}}{a^2}$$

2.2.2.2

$$e_q = \frac{a^2 - b^2}{a^4}$$

2.2.2.3

$$e_1 = \frac{1 - \sqrt{1 - e_q}}{1 + \sqrt{1 - e_q}}$$

2.2.2.4

$$e'^2 = \frac{e_q}{1 - e_q}$$

Dem Ostwert werden aufgrund des Abstandes zum Bezugsmeridian 500.000 Meter abgezogen.

2.2.2.5

$$x_{-500000} = x - 500.000$$

Anhand des Nordwertes und des verringerten Ostwertes werden die folgenden Hilfsgrößen bestimmt.

2.2.2.6

$$M = \frac{y_{UTM}}{k_0}$$

2.2.2.7

$$\mu = \frac{M}{a \cdot \left[1 - \frac{e_q}{4} - 3 \cdot \frac{e_q^2}{64} - 5 \cdot \frac{e_q^3}{256} \right]}$$

2.2.2.8

$$\phi_1 = \mu + \left[3 \cdot \frac{e_1}{2} - 27 \cdot \frac{e_1^3}{32} \right] \cdot \sin(2 \cdot \mu) + \left[21 \cdot \frac{e_1^2}{16} - 55 \cdot \frac{e_1^4}{32} \right] \cdot \sin(4 \cdot \mu) + 151 \cdot \frac{e_1^3}{96} \cdot \sin(6 \cdot \mu)$$

2.2.2.9

$$N_1 = \frac{a}{\sqrt{1 - e_q \cdot \sin(\phi_1)^2}}$$

2.2.2.10

$$T_1 = \tan(\phi_1)^2$$

2.2.2.11

$$C_1 = e_q \cdot \cos(\phi_1)^2$$

2.2.2.12

$$R_1 = a \cdot \frac{1 - e_q}{\left[1 - e_q \cdot \sin(\phi_1)^2 \right]^{1,5}}$$

2.2.2.13

$$D = \frac{x_{-500000}}{N_1 \cdot k_0}$$

Zu beachten ist, dass die Hilfsgrößen mit dem Bogenmaß berechnet wurden.

Aus den Hilfsgrößen werden die Breitengrade ϕ und die Längengrade λ wie folgt berechnet.

2.2.2.14

$$\phi = \left[\phi_1 - N_1 \cdot \frac{\tan(\phi_1)}{R_1} \cdot \left[\frac{D^2}{2} - [5 + 3 \cdot T_1 + 10 \cdot C_1 - 4 \cdot C_1^2 - 9] \cdot \frac{D^4}{24} + [61 + 90 \cdot T_1 + 298 \cdot C_1 + 45 \cdot T_1^2 - 252 \cdot e'^2 - 3 \cdot C_1^2] \cdot \frac{D^6}{720} \right] \right] \cdot \frac{180}{\pi}$$

2.2.2.15

$$\lambda = \lambda_0 \left[\frac{D - [1 + 2 \cdot T_1 + C_1] \cdot \frac{D^6}{6} + [5 - 2 \cdot C_1 + 28 \cdot T_1 - 3 \cdot C_1^2 + 8 \cdot e'^2 + 24 \cdot T_1^2] \cdot \frac{D^5}{120}}{\cos(\phi_1)} \right] \cdot \frac{180}{\pi} + 9$$

Da zuvor, anders als bei Snyder, im Bogenmaß gerechnet wurde, werden die Breitengrade ϕ und die Längengrade λ in den Gleichungen 2.2.2.14 und 2.2.2.15 mit $180/\pi$ multipliziert. Des Weiteren wird für die UTM Zone 32 dem Längengrad die Zahl 9 addiert, da diese Zone den Bezugsmeridian 9 Grad Ost hat.

3 Material und Methoden

3.1 Material

Die für diese Arbeit verwendeten Strömungsdaten stammen aus dem Strömungsmodell Operational Current Information System (OpCIS).

OpCIS wurde im Auftrag der Hamburger Port Authority (HPA) von der DHI entwickelt. Das Modell gibt Auskunft über Strömungsverhältnisse in einem Gebiet mit einer Gesamtfläche von rund 70 km², das sich vom Wehr Geesthacht bis zum Pegel Stadersand erstreckt. OpCIS berechnet die Strömungsdaten anhand von Onlinemesswerten und externen, regionalen Vorhersagen. Die Ergebnisse werden über einen Webservice bereitgestellt und liegen für sechs unterschiedliche Zoomstufen vor. Zur Verfügung steht die Strömungs-geschwindigkeit und -richtung,

sowie der nächste Kenterpunkt. Die Daten stehen für die Zeitpunkte Echtzeit und Prognose (in 30 Minuten) zu Verfügung. Eine Aktualisierung der Strömungsdaten erfolgt alle fünf Minuten. Eine visuelle Aufbereitung der Daten ermöglicht es die Strömungsverhältnisse über den OGC-konforme Web Map Services (WMS) auf einer Karte darzustellen. Eine beispielhafte Darstellung der Karte mit den Strömungsdaten ist in Abbildung 4 zu sehen. Die Strömungsrichtung wird durch die Pfeile angezeigt und die Strömungsgeschwindigkeit in Knoten an den jeweiligen Pfeilen angegeben.

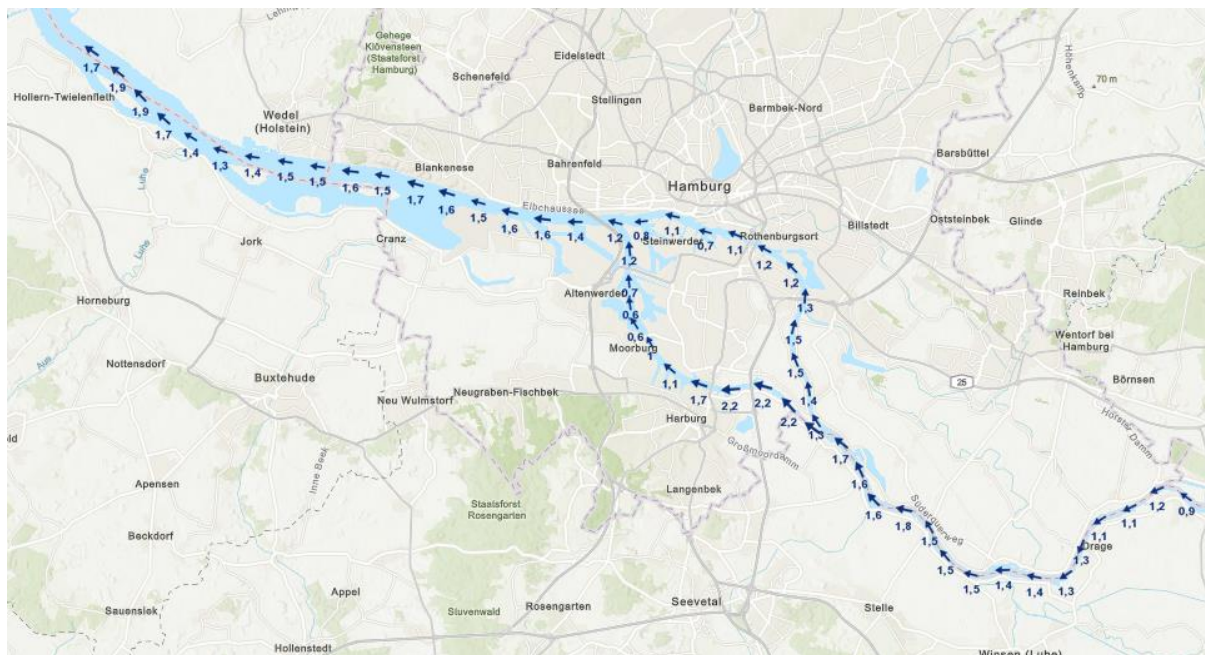


Abbildung 4 Bereich für den Strömungsdaten vorliegen

Für die Modellierung wird die DHI-Software MIKE21 FM (Flexible Mesh) genutzt. Die Software erlaubt es das Gebiet in unterschiedlich hoher räumlicher Auflösung darzustellen (11). Dabei werden Regionen mit komplexen Strömungsverhältnissen in einer höheren Auflösung dargestellt, als Regionen, die von weniger großem Interesse sind. Hierzu wurde das Modellgebiet in unterschiedlich große Polygone geteilt, innerhalb derer die Strömungsgeschwindigkeit und Strömungsrichtung gemittelt wird. Die Größe der Polygone wird den jeweiligen Zoomstufen angepasst. Die modellierten Messgrößen werden mit den drei Dauerströmungsmessanlagen des Hamburger Hafens überprüft. Ein Modelllauf wird alle 60 Minuten mit aktualisierten Messdaten durchgeführt. Jeder Lauf beinhaltet eine Modellierung der Strömungsverhältnisse, die 1,5 Stunden in der Vergangenheit liegen. Dies dient der Qualitätsüberprüfung des Modells, da so die modellierten Daten mit gemessenen Daten verglichen werden können. Anhand der Ergebnisse erfolgt eine Prognose für bis zu 8,5 Stunden in die Zukunft, um eine Erkennung der Tidenkenterung zu gewährleisten.

3.2 Methode

In den folgenden Abschnitten wird der Vorgang bei der Recherche, sowie die Methodik bei der Simulation der Bewegung einer Person durch die Strömung erläutert. Es wird die Simulation, die Bestimmung der Konstanten, die numerische Diskretisierung und die Trajektorien-Berechnung erklärt. Des Weiteren wird beschrieben, wie die Simulation umgesetzt wurde.

3.2.1 Recherche

Für die Bestimmung der Verweildauer einer Person an der Wasseroberfläche wurde eine Recherche durchgeführt. Für die Recherche wurden die Datenbanken PubMed, Springer und Google Scholar genutzt. Es wurden folgende Suchbegriffe verwendet: human body flotation; Salzgehalt Elbe; Wasserleichen; Buoyancy (deutsch: Auftrieb); Buoyancy cloth; Human body buoyancy; displacement body drowning; time drowning reach bottom; resurface drown.

3.2.2 Bestimmung der Konstanten A , c_w , M und ρ

Die Masse M der Person wird auf 80 kg gesetzt und die Dichte des Wassers ρ wird mit 1000 kg/m^3 angenommen. Für c_w wird ein Wert von 0,8 gewählt. Der Wert stammt aus einer Versuchsreihe in der Widerstandbeiwerte von unterschiedlichen Personen für die Schwimmtechniken Kraul, Brust, Delphin und Rücken gemessen wurden (12). Der Widerstandbeiwert c_w ist stark vom Geschlecht, dem Individuum und der Schwimmtechnik abhängig. Es wurde hier der höchste gemessene Widerstandbeiwert für einen männlichen Brustschwimmer gewählt. Die Projektionsfläche A wurde aus dem 2/3-Exponenten des Körpervolumens berechnet (12).

3.2.2.1

$$A = V^{\frac{2}{3}}$$

Wobei V aus dem Quotienten der Körperdichte ρ_k und M berechnet wurde. Für ρ_k wurde der in Kapitel ermittelte Wert von $1011,24 \text{ kg/m}^3$ eingesetzt. Daraus ergibt sich für A folgender Wert:

3.2.2.2

$$A = \frac{M^{\frac{2}{3}}}{\rho_k} = 0,18 \text{ m}^2$$

3.2.3 Numerische Diskretisierung

Die Ableitung nach der Zeit $\frac{dv}{dt}$ wird mittels Expliziten Euler-Verfahrens diskretisiert.

Es gilt:

3.2.3.1

$$\frac{dv}{dt} \stackrel{!}{=} f(v, t)$$

3.2.3.2

$$v(t_0) \stackrel{!}{=} 0$$

3.2.3.3

$$v_{n+1} = v_n + h \cdot f(v_n, t_n)$$

h ist die Diskretierungs-Schrittweite mit der Schrittweite $h = 1$ Sekunden.

Die Bewegung der Person zieht nicht nur eine zeitliche Änderung nach sich, sondern auch eine Ortsänderung. Da die Strömungsgeschwindigkeit Orts- und Zeitabhängig ist, wird die Strömungsgeschwindigkeit jedes Mal für den Zeitschritt n und den Ort zum Zeitschritt n neu gesetzt.

Es ergibt sich:

3.2.3.4

$$v_{n+1} = v_n + h \cdot \frac{1}{M} \cdot c_w \cdot \frac{1}{2} \cdot \rho \cdot |w_n - v_n| \cdot (w_n - v_n) \cdot A$$

3.2.4 Trajektorien Berechnung

Da Strömungsgeschwindigkeit und Strömungsrichtung sich im Verlauf der Bewegung ändern, werden für jede neue Position die Strömungsdaten abgefragt und angepasst. Die geographische Position P wird in Form der Universalen Transversalen Merkatorprojektion (UTM) für die Zone 32 angegeben (13). Zur Beschreibung der geographischen Position werden die Symbole x (geographische Breite) und y (geographische Länge) genutzt.

Da durch die UTM-Projektion die Erdoberfläche in den jeweiligen Zonen in einer ebenen Fläche abgebildet wird, kann der Einfluss der Erdkrümmung vernachlässigt werden. Die benötigten Strömungsdaten werden vom Simulationswebservice des Operational Current Information System (OpCIS) abgerufen. Die Geschwindigkeit, mit der die Person driftet bzw. sich bewegt, wird gemäß Kapitel 4.2.2 berechnet. Die geographische Position einer Person im Zeitschritt n wird wie folgt bezeichnet:

3.2.4.1

$$P_n = \begin{bmatrix} x \\ y \end{bmatrix}_n$$

Ausgehend von der Driftgeschwindigkeit v_n am Ort P_n , wird die Geschwindigkeit v_{n+1} berechnet. Der neue Ort P_{n+1} wird über eine Vektorverschiebung berechnet:

3.2.4.2

$$S = \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} v_n \cdot \Delta t \cdot \sin(\alpha_n) \\ v_n \cdot \Delta t \cdot \cos(\alpha_n) \end{bmatrix}$$

Wobei S die Strecke beschreibt, die die Person in der Zeit Δt zurückgelegt hat und α_n , der Strömungsrichtung im Punkt P_n entspricht. Somit berechnet sich die neue Position der Person durch:

3.2.4.3

$$P_{n+1} = \begin{bmatrix} x \\ y \end{bmatrix}_n + \begin{bmatrix} v_n \cdot \Delta t \cdot \sin(\alpha_n) \\ v_n \cdot \Delta t \cdot \cos(\alpha_n) \end{bmatrix}$$

In Abbildung 5 ist die Berechnung visualisiert.

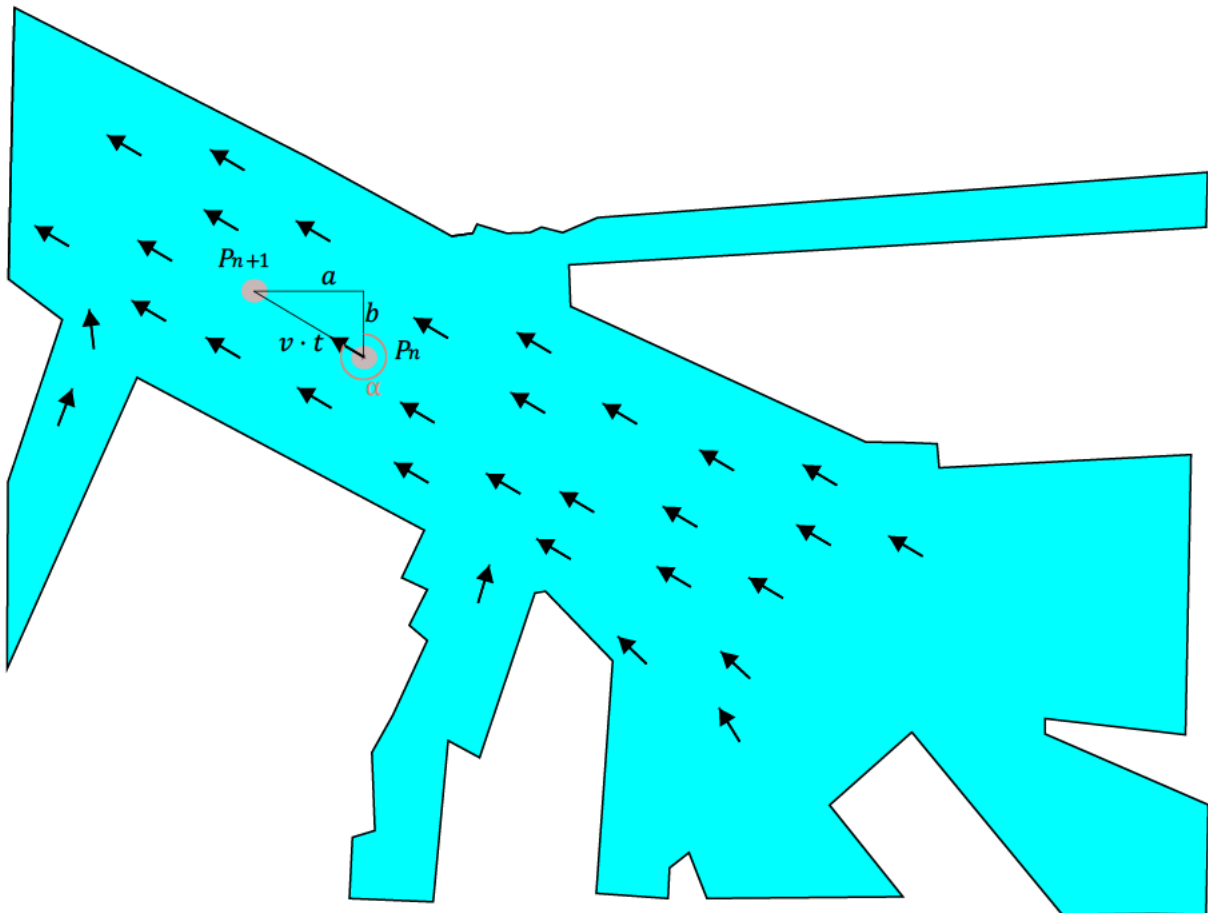


Abbildung 5 Berechnung der geographischen Position

3.2.5 Streuung des Startpunktes

Da eine metergenaue Standortangabe der Position, der Person unwahrscheinlich ist, wird der Drift nicht nur von einem Startpunkt, sondern von mehreren Startpunkten aus berechnet. Es wurde sich für fünf Startpunkte entschieden. Die Punkte sind wie in Abbildung 5 gezeigt angeordnet.

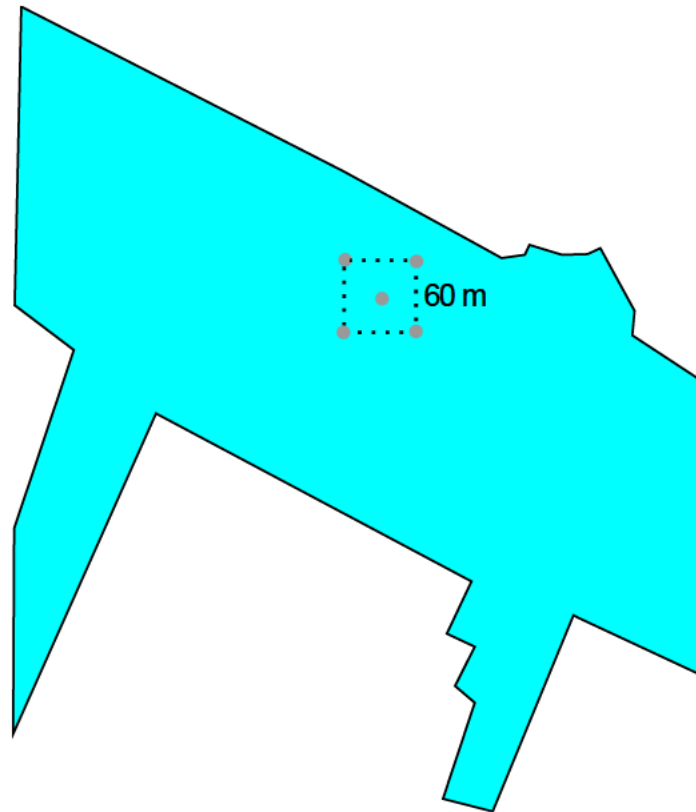


Abbildung 6 Verteilung der Startpunkt für die Simulation

Die äußeren 4 Punkte werden aus dem mittleren Punkt berechnet. Sie haben alle den gleichen Abstand zum Mittelpunkt und jeweils 60 Meter Abstand zu den beiden nächsten äußeren Punkten.

3.2.6 Umsetzung in C#

Für die Modellierung der Bewegung einer Person in der Elbe wurde das Modell in C# geschrieben und berechnet. C# ist eine objektorientierte Programmiersprache. Als Entwicklungsumgebung wurde Visual Studio genutzt. Das Modell simuliert die Bewegung einer Person durch die Strömung für eine Dauer von 30 Minuten. Die Simulation wird gemäß Kapitel 3.2.5. von mehreren Startpunkten gleichzeitig durchgeführt. Nach Beendigung der Berechnungen öffnet das Programm den Online-Kartendienst Geoplaner, in dem die Bewegung visualisiert wird und speichert die berechneten Koordinaten in einer Excel-Datei.

Im Verlaufe der Fertigstellung des Modells wurden zahlreiche Simulationen durchgeführt. Die erste Simulation wurde mit einem in Excel geschriebenen

Programm ausgeführt. Dabei waren Simulationen nur für einen Punkt möglich. Des Weiteren wurde hier vernachlässigt, dass ein Mensch Aufgrund des Widerstandbeiwertes mit einer geringeren Geschwindigkeit als die Strömungsgeschwindigkeit bewegt wird. Da Excel eine relativ lange Zeit für die Auswertung der Daten benötigte, musste die Anzahl an Schritten möglichst gering gehalten werden. Dies führte zur Wahl einer Schrittweite von mehreren Sekunden. Zudem kam es Aufgrund von Verbindungsfehlern regelmäßig zu Abbrüchen der Simulation. Aus diesem Grund wurde das Modell auf C# umgeschrieben. Dies machte es möglich die Simulation von mehreren Punkten aus gleichzeitig ablaufen zu lassen. Zudem wurde die Zeit für die Berechnungen deutlich verkürzt, was eine Schrittweite von einer Sekunde möglich machte. Da sich auch hier zeigte, dass Verbindungsfehler auftraten, wurden hierfür Ausnahmen eingeführt. Dies machte das Programm deutlich weniger fehleranfällig. Im letzten Schritt wurde noch der Widerstandbeiwert für die Modellierung hinzugefügt.

Das geschriebene Programm besteht aus zwei Klassen. Die Klasse „Koordinaten“ beinhaltet die statische Hauptmethode „Main“. Nach dem Start des Programms werden die Koordinaten für den Startpunkt der Simulation händisch in die Konsole eingetragen. Die x und y Koordinaten werden im UTM Format eingegeben. Da die in der Konsole eingegebenen Koordinaten vom Datentyp String sind, werden diese in den Datentyp Double konvertiert und gespeichert. Die konvertierten Daten werden an die Klasse „Programm“ weitergegeben. Die Klasse „Programm“ ist für die eigentliche Berechnung der Simulation und dem Speichern der Daten in Excel zuständig. Nach der Übergabe der Daten, wird die Methode „mainthread“ der Klasse „Programm“ gestartet.

```

class Koordinaten
{
    static void Main(string[] args)
    {
        string Eingabexstartstr = Console.ReadLine();
        Console.WriteLine(Eingabexstartstr);

        double Eingabexstart = Convert.ToDouble(Eingabexstartstr);

        string Eingabeystartstr = Console.ReadLine();
        Console.WriteLine(Eingabeystartstr);

        double Eingabeystart = Convert.ToDouble(Eingabeystartstr);

        Programm start = new Programm() { Eingabex = Eingabexstart, Eingabey =
        Eingabeystart };

        start.mainthrad();
    }
}

```

Abbildung 7 Programmcode

Die Methode „mainthrad“ initialisiert die Methoden „thread1“, „thread2“, „thread3“, „thread4“ und „thread5“ so, dass alle 5 Methoden gleichzeitig ablaufen.

```

class Programm
{
    public double Eingabex { get; set; }
    public double Eingabey { get; set; }

    public void mainthrad()
    {
        var a = new Task(thread1);
        var b = new Task(thread2);
        var c = new Task(thread3);
        var d = new Task(thread4);
        var e = new Task(thread5);

        a.Start();
        b.Start();
        c.Start();
        d.Start();
        e.Start();
    }
}

```

Abbildung 8 Programmcode

Erst nach dem die Methoden beendet wurden, werden die weiteren Methoden „Umrechnung1“, „Umrechnung2“, „Umrechnung3“, „Umrechnung4“ und Umrechnung 5“ sowie die Methoden „Web“, „Excel“ und „Weg“ nacheinander initialisiert.

```

Task.WaitAll(a, b,c,d,e);

    Umrechnung1();
    Umrechnung2();
    Umrechnung3();
    Umrechnung4();
    Umrechnung5();

    Web();
    Excel();
    weg();

}

```

Abbildung 9 Programmcode

In den Methoden „thread1“ bis „thread5“ findet die Trajektorien Berechnung statt. Jede Methode berechnet einen der fünf Bewegungsabläufe. In „thread1“ wird die Berechnung für den händisch in die Konsole eingegebenen Punkt durchgeführt. Während in den übrigen Methoden die Berechnung für die nach Kapitel 3.2.5 berechneten Punkte durchgeführt wird. Die Koordinaten, Geschwindigkeit und Strecke werden für jede Methode in eigenen Arrays gespeichert.

```

public double[] xarray1 = new double[1800];
public double[] yarray1 = new double[1800];
public double[] speed1 = new double[1800];
public double strecke1 = 0;

public void thread1() ...

public double[] xarray2 = new double[1800];
public double[] yarray2 = new double[1800];
public double[] speed2 = new double[1800];
public double strecke2 = 0;

public void thread2() ...

public double[] xarray3 = new double[1800];
public double[] yarray3 = new double[1800];
public double[] speed3 = new double[1800];
public double strecke3 = 0;

public void thread3() ...

public double[] xarray4 = new double[1800];
public double[] yarray4 = new double[1800];
public double[] speed4 = new double[1800];
double strecke4 =0;

public void thread4()...

public double[] xarray5 = new double[1800];
public double[] yarray5 = new double[1800];
public double[] speed5 = new double[1800];
public double strecke5 = 0;

public void thread5()...

```

Abbildung 10 Programmcode

Die Umsetzung der Berechnung wird am Beispiel von „thread1“ vorgestellt. Zunächst werden die Variablen festgelegt. Die Variablen „eingabex“ und „eingabey“ entsprechen den Koordinaten x und y im UTM Format, für die die Strömungsdaten gesucht werden. Die Variable „eingabex5“ entspricht der Verschiebung der x-Koordinate um 5 m und die Variable „eingabey5“ der Verschiebung der y-Koordinate um 5 m. Zusammen bilden die Variablen „eingabex“, „eingabey“, „eingabex5“ und „eingabey5“ eine Bounding Box (BBox). Die BBox beschreibt einen Begrenzungsrahmen, der durch die beiden Längengrade „eingabey“, „eingabey5“ und den Breitengraden „eingabex“ und „eingabex5“ definiert wird. Die Seitenlänge des Rahmens beträgt 5 Meter. Für die BBox werden die Strömungsdaten im OpCIS Webdienst abgefragt. Alle übrigen Variablen werden im Verlauf des Kapitels vorgestellt.


```

public void thread1()
{
    double riverdirection = 0;
    double riverspeed = 0;
    double speed = 0;
    double speedalt = 0;
    double alpha = 0;
    double a = 0;
    double b = 0;
    double x = 0;
    double y = 0;
    double eingabex;
    double eingabey;
    double eingabex5;
    double eingabey5;
    eingabey = Eingabey;
    eingabex = Eingabex;
    x = Eingabex;
    y = Eingabey;
    eingabex5 = eingabex + 5;
    eingabey5 = eingabey + 5;
    double j = 0;
}

```

Abbildung 12 Programmcode

Zur Bestimmung der Dauer eines Ablaufes wird die Klasse „Stopwatch“ verwendet.

```

Stopwatch stopWatch = new Stopwatch();
stopWatch.Start();
int t = 0;
int talt = 0;

```

Abbildung 11 Programmcode

Die Berechnung wird in einer „For-Schleife“ mit 1800 Durchgängen genutzt. Jeder Durchgang entspricht einer Schrittweite von einer Sekunde. Wodurch eine Gesamtdauer von 30 Minuten erreicht wird. Zu Beginn der Schleife wird die Variable „URLString“ festgelegt. Die Variable „URLString“ besteht aus einem am Anfang stehenden Textteil, der sich aus den ersten Zeichenfolgen der URL zusammensetzt. Daraufhin folgt die BBox, mit den Koordinaten, die in jedem Durchlauf aktualisiert werden. Am Ende des Strings steht der letzte Teil der URL.

```

for (int i = 0; i < 1800; i++)
{
    String urlString = Convert.ToString(
        "https://opcis.hpanet.de/arcgis/services/opcis_civilpro/OpCIS_Echtzeit/MapServer/WmsServer?Request=GetFeatureInfo&VERSION=1.3.0&SERVICE=WMS&CRS=EPSG:25832&BBOX="
        + eingabex + "," +
        eingabey + "," + eingabex5 + "," + eingabey5 +
        "_____");
    XmlTextReader reader = new XmlTextReader(urlString);
}

```

Abbildung 13 Programmcode

Mithilfe der Klasse „XmlTextReader“ wird nun die Strömungsgeschwindigkeit aus dem Webservice gelesen und in den Datentyp Double konvertiert. Die Daten werden in der Variablen „riverspeed“ gespeichert. Für den Fall, dass eine Verbindung mit dem Webservice fehlschlagen sollte, wird in der Konsole eine Warnung ausgegeben und statt der Geschwindigkeit die Ziffer 404 hinterlegt. Des Weiteren springt die Schleife einen Schritt weiter.

```
try
{
    while (reader.Read())
    {
        if (reader.HasAttributes)
        {
            string riverspeedstr = reader.GetAttribute("speed");
            riverspeed = Convert.ToDouble(riverspeedstr);
        }
    }
}
catch
{
    Console.WriteLine("in Verbindungsversuch ist fehlgeschlagen, da die Gegenstelle nach einer bestimmten Zeitspanne nicht richtig reagiert hat, oder die hergestellte Verbindung war fehlerhaft, da der verbundene Host nicht reagiert hat.");
    speed1[i] = 404;
    i = i+1;
}
```

Abbildung 14 Programmcode

Die Abfrage der Daten für die Strömungsrichtung läuft wie die Abfrage der Daten der Strömungsgeschwindigkeit ab. Wobei die Strömungsrichtung in der Variablen „riverspeed“ gespeichert wird.

```

XmlTextReader reader2 = new XmlTextReader(URLString);
try
{
    while (reader2.Read())
    {

        if (reader2.HasAttributes)
        {
            string riverdirectionstr = reader2.GetAttribute("dir");
            riverdirection = Convert.ToDouble(riverdirectionstr);

        }
    }
}
catch
{
    Console.WriteLine("in Verbindungsversuch ist fehlgeschlagen,
da die Gegenstelle nach einer bestimmten Zeitspanne nicht
richtig reagiert hat, oder die hergestellte Verbindung war
fehlerhaft, da der verbundene Host nicht reagiert hat.");
    speed1[i] = 404;
    i = i +1;
}

```

Abbildung 15 Programmcode

Nach Einholung der Strömungsdaten wird die Trajektorien Berechnung gemäß dem Kapitel 3.2.4 durchgeführt. Der Betrag wird durch eine If-Anweisung überprüft. Die Geschwindigkeit, in der die Person sich fortbewegt, wird in der Variablen „speed“ hinterlegt. Die Variable „speedalt“ entspricht der Geschwindigkeit der Person in einer zurückliegenden Schrittweite und wird nach der Berechnung der aktuellen Geschwindigkeit aktualisiert. Die Strömungsrichtung wird im Bogenmaß in der Variablen „alpha“ gespeichert. Hiermit werden die Vektorverschiebungen berechnet und in den Variablen „a“ und „b“ gespeichert. Aus der Verschiebung werden die neuen Koordinaten berechnet und in den Variablen „x“ und „y“ gespeichert. Da für die URL nur ganze Zahlen genutzt werden können, werden die Koordinaten als ganze Zahlen in den Variablen „eingabex“ und „eingabey“ gespeichert. Danach werden auch die Variablen „eingabex5“ und „eingabey5“ aktualisiert.

```

if (((riverspeed / 1.944) - speedalt) > 0)
{
    speed = speedalt + 0.824 * 0.5 * 1 / 80 * 1000 * 0.18 *
        ((riverspeed / 1.944) - speedalt) * ((riverspeed / 1.944) -
        speedalt);
}
else
{
    speed = speedalt + 0.824 * 0.5 * 1 / 80 * 1000 * 0.18 *
        ((riverspeed / 1.944) - speedalt) * (-1) * ((riverspeed /
        1.944) - speedalt);
}
speedalt = speed;
alpha = riverdirection * Math.PI / 180;
a = Math.Sin(alpha) * speed;
b = Math.Cos(alpha) * speed;

x = x + a;
y = y + b;

eingabex = x - x % 1;
eingabey = y - y % 1;
eingabex5 = eingabex + 5;
eingabey5 = eingabey + 5;

```

Abbildung 16 Programmcode

Die berechneten Koordinaten werden als ganze Zahl in den Arrays „xarray1“ und „yarray1“ gespeichert. Die Geschwindigkeit wird im Array „speed1“ hinterlegt.

```

xarray1[i] = eingabex;
yarray1[i] = eingabey;
speed1[i] = speed;

```

Abbildung 17 Programmcode

In einer If-Anweisung wird alle 300 Durchgänge die Dauer der letzten 300 Durchgänge gemessen und die Schleife für 5 Minuten abzüglich der Dauer der letzten 300 Durchgänge angehalten. Da 300 Durchgänge die Bewegung in 5 Minuten simulieren und die Strömungsdaten alle 5 Minuten aktualisiert werden, können so immer die aktuellen Strömungsdaten für die Rechnung genutzt werden.

```

if (j >= 300)
{
    j = 0;

    talt = t;
    stopwatch.Stop();
    TimeSpan tsm = stopwatch.Elapsed;
    int ts = int.Parse(tsm.Seconds.ToString());
    int tm = int.Parse(tsm.Minutes.ToString());
    t = (tm * 60 + ts) *1000;

    Thread.Sleep(300000 - t + talt);

    stopwatch.Start();

}
j = j + 1;

```

Abbildung 18 Programmcode

In der Konsole wird die Uhrzeit, die Zahl der Durchläufe, die Koordinaten und die Geschwindigkeit für jeden Durchlauf ausgegeben. Am Ende der Schleife wird die

```

Console.WriteLine(DateTime.Now);
Console.WriteLine("a" + i + " x" + xarray1[i] + " y" +
yarray1[i] + " geschw.:" + speed1[i]);

strecke1 = strecke1 + speed;

```

Abbildung 19 Programmcode

Strecke berechnet, die insgesamt von der Person zurückgelegt wurde und in der Variablen „strecke1“ gespeichert.

In den Methoden „Umrechnung1“, „Umrechnung2“, „Umrechnung3“, „Umrechnung4“ und „Umrechnung 5“ werden jeweils die UTM Koordinaten der zuvor beendeten Methoden „thread1“ bis „thread5“ in Breiten- und Längengrade umgerechnet. Die umgerechneten Koordinaten werden für jede Methode in Arrays gespeichert.

```

double[] xarraygps1 = new double[1800];
double[] yarraygps1 = new double[1800];
public void Umrechnung1()...

double[] xarraygps2 = new double[1800];
double[] yarraygps2 = new double[1800];
public void Umrechnung2()...

double[] xarraygps3 = new double[1800];
double[] yarraygps3 = new double[1800];

public void Umrechnung3()...
double[] xarraygps4 = new double[1800];
double[] yarraygps4 = new double[1800];

public void Umrechnung4()...

double[] xarraygps5 = new double[1800];
double[] yarraygps5 = new double[1800];

public void Umrechnung5()...

```

Abbildung 16 Programmcode

Die Umrechnung verläuft gemäß Kapitel 2.3 und wird für jede Koordinate in Form einer Schleife umgerechnet. Die Umsetzung wird am Beispiel der Methode „Umrechnung1“ vorgestellt. Zunächst werden die Variablen für die Umrechnung

```

public void Umrechnung1()
{
    double aa;
    double bb;
    double k0;
    double e;
    double eq;
    double e1;
    double estrich2;
    double N1;
    double T1;
    double C1;
    double R1;
    double D;
    double M;
    double mu;
    double Phi;
    double Phirad;
    double x;

```

Abbildung 17 Programmcode

festgelegt. Die Bezeichnung der Variablen ähnelt den Bezeichnungen in Kapitel 2.3.

Danach folgt die Festlegung, der für den Raum Hamburg geltenden und in Kapitel 2.3 bestimmten Hilfsgrößen.

```

aa = 6378137;
bb = 6356752.31424518;
k0 = 0.9996;
e = Math.Sqrt((aa * aa - bb * bb) / (aa * aa));
eq = (aa * aa - bb * bb) / (aa * aa);
e1 = (1 - Math.Sqrt(1 - eq)) / (1 + Math.Sqrt(1 - eq));

```

Abbildung 18 Programmcode

In einer Schleife werden nun alle zuvor berechneten Koordinaten in geographische Kugelkoordinaten umgerechnet. Die Breitengrade werden im Array „xarraygps1“ und die Längengrade im Array „yarraygps2“ in Grad gespeichert. Die Koordinaten

```

for (int i = 0; i < 1800; i++)
{
    x = xarray1[i] - 500000;
    estrich2 = eq / (1 - eq);
    M = yarray1[i] / k0;
    mu = M / (aa * (1 - eq / 4 - 3 * eq * eq / 64 - 5 * eq * eq * eq /
        256));
    Phirad = mu + (3 * e1 / 2 - 27 * e1 * e1 * e1 / 32) * Math.Sin(2 *
        mu) + (21 * e1 * e1 / 16 - 55 * e1 * e1 * e1 * e1 / 32) *
        Math.Sin(4 * mu) + (151 * e1 * e1 * e1 / 96) * Math.Sin(6 *
        mu);
    Phi = Phirad * 180 / Math.PI;
    e1 = (1 - Math.Sqrt(1 - eq)) / (1 + Math.Sqrt(1 - eq));
    N1 = aa / Math.Sqrt(1 - eq * Math.Sin(Phirad) * Math.Sin(Phirad));
    T1 = Math.Tan(Phirad) * Math.Tan(Phirad);
    C1 = eq * Math.Cos(Phirad) * Math.Cos(Phirad);
    R1 = aa * (1 - eq) / Math.Pow((1 - eq * Math.Sin(Phirad) *
        Math.Sin(Phirad)), 1.5);
    D = x / (N1 * k0);

    xarraygps1[i] = Phirad - (N1 * Math.Tan(Phirad) / R1)
        * (D * D / 2 - (5 + 3 * T1 + 10 * C1 - 4 * C1 * C1 - 9 *
            estrich2) * D * D * D * D / 24 + (61 + 90 * T1 + 298 * C1 + 45
            * T1 * T1 - 252 * estrich2 - 3 * C1 * C1) * D * D * D * D *
            D / 720);
    xarraygps1[i] = xarraygps1[i] * 180 / Math.PI;

    yarraygps1[i] = (D - (1 + 2 * T1 + C1) * D * D * D / 6 +
        (5 - 2 * C1 + 28 * T1 - 3 * C1 * C1 + 8 * estrich2 + 24 * T1 *
            T1) * D * D * D * D * D / 120) / Math.Cos(Phirad);

    yarraygps1[i] = yarraygps1[i] * 180 / Math.PI + 9;

    Console.WriteLine(xarraygps1[i] + " " + yarraygps1[i]);
}

```

Abbildung 19 Programmcode

werden in der Konsole ausgegeben.

In der Methode „Weg“ wird für jeden Bewegungsablauf jeweils der Abstand jedes Punktes zum Startpunkt berechnet. Für jeden Bewegungsablauf wird in einer For-Schleife der Abstand eines Punktes zu seinem Vorgänger bestimmt. Diesem Abstand

wird der Abstand des Vorgängers zum Startpunkt addiert. Es ergibt sich der Abstand des Punktes zum Startpunkt. Die Abstände werden in einem Array gespeichert.

```
public void weg()
{
    weg1[0] = 0;
    for (int i = 1; i < 1800; i++)

        {

            weg1[i] = weg1[i - 1] + speed1[i-1];

        }
    weg2[0] = 0;
    for (int i = 1; i < 1800; i++)...
    weg3[0] = 0;
    for (int i = 1; i < 1800; i++)...
    weg4[0] = 0;
    for (int i = 1; i < 1800; i++)...
    weg5[0] = 0;
    for (int i = 1; i < 1800; i++)...
```

Abbildung 20 Programmcode

Die Methode „Web“ öffnet im Webbrowser einen Online-Kartendienst, in dem die Bewegung visualisiert wird. Zunächst werden die Variablen „Koordinaten“ und „Koordinatenalt“ eingeführt, in denen die Koordinaten im Dateityp String gespeichert werden.

```
public void Web()
{

    string Koordinaten;
    string Koordinatenalt;
    Koordinaten = null;

    int j = 0;
```

Abbildung 21 Programmcode

Es folgt eine For-Schleife mit 1800 Durchläufen, in der die Koordinaten der Methode „Umrechnung1“ in den Dateityp String konvertiert werden. Des Weiteren werden Kommas durch Punkte ersetzt.

```
for (int i = 0; i < 1800; i++)
{
    string ein = Convert.ToString(xarraygps1[i],
    CultureInfo.InvariantCulture);
    string zwei = Convert.ToString(yarraygps1[i],
    CultureInfo.InvariantCulture);
```

Abbildung 22 Programmcode

In einer If-Anweisung werden das erste Koordinatenpaar, sowie das 300ste, 600ste, 900ste, 1200ste, 1500ste und das letzte Koordinatenpaar in einem einzelnen String in der Variablen „Koordinaten“ gespeichert. Die Koordinaten müssen in einer speziellen Form gespeichert werden, damit die Punkte von der Webkarte „Geoplaner“ erkannt und im Webbrowser geöffnet werden können. Am Anfang jedes Punktes steht die Zeichenfolge „p=“. Daraufhin folgt der Breitengrad und der Längengrad. Die beiden Grade werden durch ein Komma getrennt. Am Ende der Koordinaten steht ein Semikolon. So würde die Variable „Koordinaten“, in der zwei Koordinaten gespeichert wurden, folgenden Inhalt haben:

```
p=53.54407617481908,9.908276236842559;p=53.54407617481908,9.908276236842559;
```

```
if (Koordinaten == null)
{
    Koordinatenalt = "p=" + ein + "," + zwei + ";";
    Koordinaten = Koordinatenalt;
}
else
{
    if( j == 300)
    {
        Koordinatenalt = Koordinaten + "p=" + ein + "," + zwei +
        ";";
        Koordinaten = Koordinatenalt;

        j = 0;
    }
}
j = j + 1;
if (i == 1799)
{
    Koordinatenalt = Koordinaten + "p=" + ein + "," + zwei + ";";
    Koordinaten = Koordinatenalt;
}
}
```

Abbildung 23 Programmcode

Die Koordinaten der Methoden „Umrechnung2“ bis „Umrechnung5“ werden auf gleiche Weise nacheinander in der Variablen „Koordinaten“ gespeichert. In der Variablen „link“ wird die URL gespeichert, um die Webkarte Geoplaner

(<https://www.geoplaner.de/>) mit den berechneten Koordinaten zu öffnen. Mit Klasse „Process“ wird im Webbrowser die Karte in Geoplaner geöffnet und die Bewegung abgebildet.

```

string link = "/c start https://www.geoplaner.de/?z=14;m=de;" +
Koordinaten + ";";

Console.WriteLine(link);

System.Diagnostics.Process.Start("cmd.Exe", link);
}

```

Abbildung 24 Programmcode

Ein Beispiel für eine im Webbrowser geöffnete Karte wird Abbildung 25 dargestellt. Wobei die Punkte A, H, O, V und # (bzw. 29) die jeweiligen Startpunkte sind.

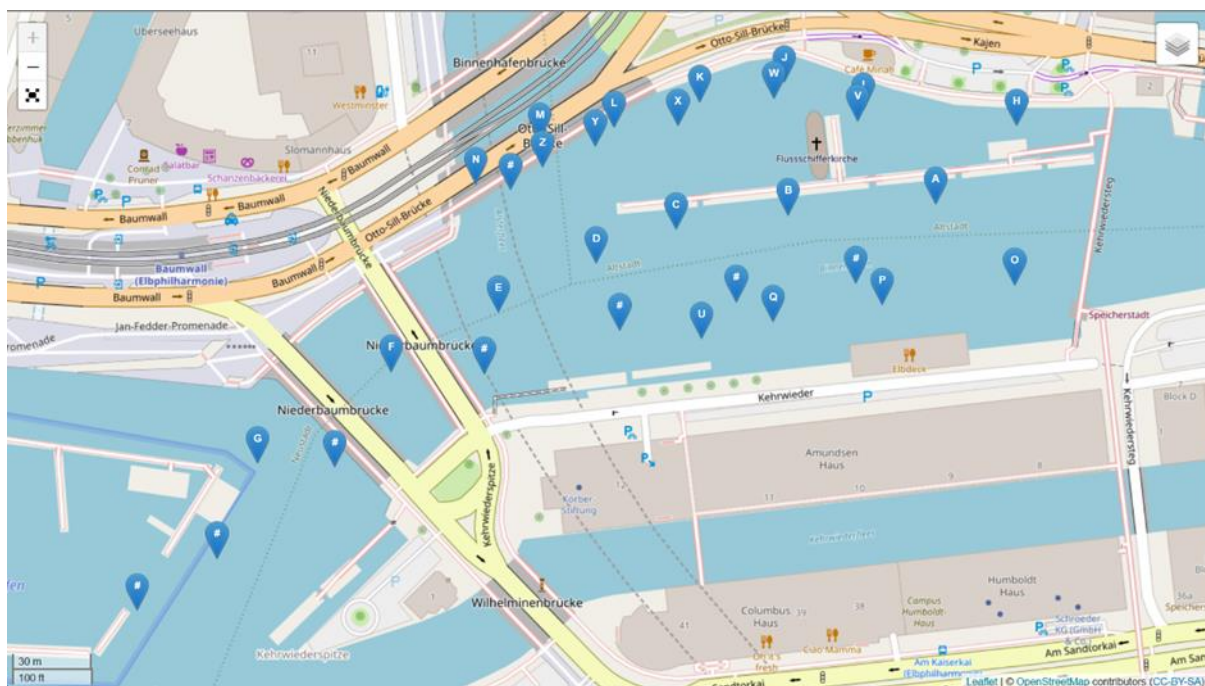


Abbildung 25 im Webbrowser geöffnete Karte

Die Methode „Excel“ gibt alle berechneten Koordinaten im UTM Format sowie die Strömungsgeschwindigkeit in den jeweiligen Punkten und die Abstände der jeweiligen Punkte zum Startpunkt einer Excel Datei aus. Hierzu wird eine Exceldatei und der Name für die Datei festgelegt.

```
public void Excel()
{
    ExcelPackage.LicenseContext = LicenseContext.NonCommercial;

    string teil1dateiname =
"C:\\Users\\manue\\OneDrive\\Documents\\bachelorarbeit\\simulation\\c\\Kohlbrand";
    int teild2dateiname = 1;
    string dateiname = Convert.ToString(teil1dateiname + teild2dateiname +
".xlsx");
}
```

Abbildung 26 Programmcode

In einer While-Anweisung wird die Formatierung der Exceltabelle bestimmt und die Datei gespeichert.

```

while (File.Exists(dateiname) == true)
{
    teild2dateiname = teild2dateiname + 1;

    dateiname = Convert.ToString(teil1dateiname + teild2dateiname +
        ".xlsx");
}
var file = new FileInfo(dateiname);

using var package = new ExcelPackage(file);

var ws = package.Workbook.Worksheets.Add("MainReport");
ws.Cells["A1"].LoadFromText("x");
ws.Cells["B1"].LoadFromText("y");
ws.Cells["C1"].LoadFromText("Geschwindigkeit");
ws.Cells["D1"].LoadFromText("Strecke");
ws.Cells["E1"].LoadFromText("x + 15");
ws.Cells["F1"].LoadFromText("y + 15");
ws.Cells["G1"].LoadFromText("Geschwindigkeit");
ws.Cells["H1"].LoadFromText("Strecke");
ws.Cells["I1"].LoadFromText("x + 30");
ws.Cells["J1"].LoadFromText("y + 30");
ws.Cells["K1"].LoadFromText("Geschwindigkeit");
ws.Cells["L1"].LoadFromText("Strecke");
ws.Cells["M1"].LoadFromText("x -30");
ws.Cells["N1"].LoadFromText("y -30");
ws.Cells["O1"].LoadFromText("Geschwindigkeit");
ws.Cells["P1"].LoadFromText("Strecke");
ws.Cells["Q1"].LoadFromText("x -15");
ws.Cells["R1"].LoadFromText("y -15");
ws.Cells["S1"].LoadFromText("Geschwindigkeit");

var rangex1 = ws.Cells["A2"].LoadFromCollection(xarray1, true);
rangex1.AutoFitColumns();

var rangey1 = ws.Cells["B2"].LoadFromCollection(yarray1, true);
rangey1.AutoFitColumns();

var rangey6 = ws.Cells["C2"].LoadFromCollection(speed1, true);
rangey6.AutoFitColumns();

var rangey11 = ws.Cells["D2"].LoadFromCollection(weg1, true);
rangey11.AutoFitColumns();

var rangex2 = ws.Cells["E2"].LoadFromCollection(xarray2, true);
rangex2.AutoFitColumns();

var rangey2 = ws.Cells["F2"].LoadFromCollection(yarray2, true);
rangey2.AutoFitColumns();

```

Abbildung 27 Programmcode

```

var rangey7 = ws.Cells["F2"].LoadFromCollection(speed2, true);
rangey7.AutoFitColumns();

var rangey12 = ws.Cells["G2"].LoadFromCollection(weg2, true);
rangey12.AutoFitColumns();

var rangex3 = ws.Cells["H2"].LoadFromCollection(xarray3, true);
rangex3.AutoFitColumns();

var rangey3 = ws.Cells["I2"].LoadFromCollection(yarray3, true);
rangey3.AutoFitColumns();

var rangey8 = ws.Cells["J2"].LoadFromCollection(speed3, true);
rangey8.AutoFitColumns();

var rangey13 = ws.Cells["K2"].LoadFromCollection(weg3, true);
rangey13.AutoFitColumns();

var rangex4 = ws.Cells["L2"].LoadFromCollection(xarray4, true);
rangex4.AutoFitColumns();

var rangey4 = ws.Cells["M2"].LoadFromCollection(yarray4, true);
rangey4.AutoFitColumns();

var rangey9 = ws.Cells["N2"].LoadFromCollection(speed4, true);
rangey9.AutoFitColumns();

var rangey14 = ws.Cells["O2"].LoadFromCollection(weg4, true);
rangey14.AutoFitColumns();

var rangex5 = ws.Cells["P2"].LoadFromCollection(xarray5, true);
rangex5.AutoFitColumns();

var rangey5 = ws.Cells["Q2"].LoadFromCollection(yarray5, true);
rangey5.AutoFitColumns();

var rangey10 = ws.Cells["R2"].LoadFromCollection(speed5, true);
rangey10.AutoFitColumns();

var rangey15 = ws.Cells["S2"].LoadFromCollection(weg5, true);
rangey15.AutoFitColumns();

ws.Cells["T1"].LoadFromText("Strecke");
ws.Cells["T2"].LoadFromText("Strecke");
ws.Cells["T3"].LoadFromText("Strecke");
ws.Cells["T4"].LoadFromText("Strecke");
ws.Cells["T5"].LoadFromText("Strecke");

ws.Cells["U1"].FillNumber(strecke1);
ws.Cells["U2"].FillNumber(strecke2);
ws.Cells["U3"].FillNumber(strecke3);
ws.Cells["U4"].FillNumber(strecke4);
ws.Cells["U5"].FillNumber(strecke5);

package.SaveAsync();

```

Abbildung 28 Programmcode

4 Ergebnisse

Die nachfolgenden Abschnitte 4.1 und 4.2 beschäftigen sich mit der Auswertung der Recherchearbeit und der Auswertung der Simulationen.

4.1 Der Menschliche Körper auf und unter Wasser

4.1.1 Schwimmfähigkeit des menschlichen Körpers

Wie in Kapitel 2.2.2 gezeigt, sinkt ein Körper dann, wenn die Dichte des Körpers größer als die Dichte der Flüssigkeit ist. Für diese Arbeit wird die Dichte der Flüssigkeit zur Vereinfachung auf den Wert von 1000 kg/m^3 gesetzt. Dies entspricht dem Wert von destilliertem Wasser bei $4 \text{ }^\circ\text{C}$ (14). Die Dichte des Körpers ist von Person zu Person verschieden. Grundsätzlich kann der menschliche Körper modellhaft in drei Komponenten geteilt werden: Die Fettmasse, die Magermasse und dem Knochenmineralgehalt. Knochen haben mit einer Dichte zwischen 1400 kg/m^3 und 1800 kg/m^3 die höchste Dichte (14). Die Muskeln, Sehnen und Bänder haben eine Dichte zwischen 1020 kg/m^3 und 1050 kg/m^3 (14). Die geringste Dichte weist die Fettmasse auf, mit einer Dichte zwischen 940 kg/m^3 und 950 kg/m^3 (14). Die prozentuale Verteilung der Komponenten ist individuell verschieden. In einer 2011 bis 2019 durchgeführten österreichischen Studie wurden an 10.894 Menschen im Alter von 18 bis 81 Jahren die Menge an Fettmasse und Magermasse gemessen (15). Dabei hatten Männer einen durchschnittlichen Fettmassenanteil von 29,0 % und einen durchschnittlichen Magermassenanteil von 67,2 % des Gesamtgewichtes (15). Daraus ergibt sich ein Knochenmineralgehaltsanteil von 3,8 %, was einer Gesamtdichte zwischen $1011,24 \text{ kg/m}^3$ bis $1049,5 \text{ kg/m}^3$ entspricht. Frauen hatten einen durchschnittlichen Fettmassenanteil von 38,3 % und einen durchschnittlichen Magermassenanteil von 58,8 % vom Gesamtgewicht. Daraus ergibt sich ein Knochenmineralgehaltsanteil von 2,9 % bei Frauen. Die resultierende Dichte entspricht $1000,38 \text{ kg/m}^3$ - $1033,45 \text{ kg/m}^3$. Sowohl bei Frauen als auch bei Männern ist die Dichte des Körpers größer als die des Wassers. Jedoch wurde bei dieser Berechnung nicht die Luft einberechnet, die beim Einatmen in den Körper gelangt. Bei einer normalen Atmung dringen zwischen 4-5 Liter Luft in den Körper ein (14). Dadurch wird das Volumen des Körpers erhöht, während die Masse des Körpers nur im geringen Maße ansteigt. Des Weiteren ist zu bedenken, dass es sich bei der Dichteberechnung nur um Durchschnittswerte handelt. So zeigte eine, von der Universität von Valencia durchgeführte Studie, dass 100 % der Studienteilnehmer, bei vollständiger Einatmung in der Lage waren zu schwimmen, während 99 % der Männer und 91,8 % der Frauen beim Ausatmen untergingen (14). Ein weiterer Faktor, der für den Auftrieb von Personen eine wichtige Rolle spielt, ist die Bekleidung. So hat eine Studie der University of Portsmouth gezeigt, dass beim Eintauchen mit Kleidung eine signifikante Menge an Luft zwischen den Kleidungsschichten

eingeschlossen bleibt (16). Die in den Kleidungsstücken eingeschlossene Luft sorgt für zusätzlichen Auftrieb und verzögert den Absinkvorgang (3).

4.1.2 Einfluss der Bodenreibung auf die Bewegung

Wie in Kapitel 2.2.1 dargelegt, wirken auf einen ruhenden, freischwimmenden Körper Widerstandkräfte. Diese Widerstandkräfte werden durch die Strömung erzeugt und wirken wie eine Druckkraft auf den Körper, die den ruhenden Körper in Strömungsrichtung bewegt. Auf einen Körper, der den Grund berührt, wirkt zusätzlich zur Druckkraft der Strömung noch die Reibungskraft. Die Reibungskraft entsteht durch die Berührung mit dem Grund und wirkt der Druckkraft entgegen, siehe Abbildung 2.

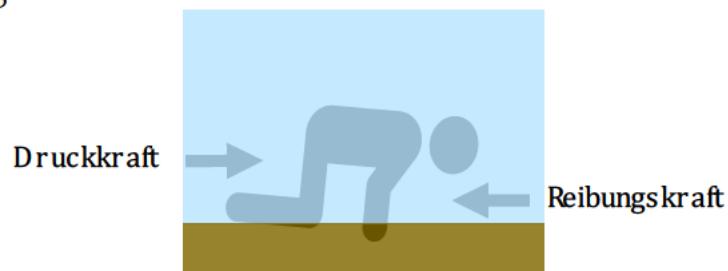


Abbildung 29: Auf den Grundgesunkene Person

Bei zu großen Reibungskräften kann der Körper nicht von der Strömung bewegt werden. So zeigte eine Studie, dass selbst Strömungen mit einer Geschwindigkeit von 0,6 m/s nicht ausreichte, um Dummies, unabhängig von ihrem Gewicht, zu bewegen (17). Eine weitere Studie zeigte, dass im Falle eines 8 Jahre alten Mädchens Küstenströme mit einer Geschwindigkeit von 1 m/s scheinbar nicht ausreichten, um den Körper über längere Strecken am Boden entlang zu bewegen (18). Hinzu kommt, dass Objekte am Boden des Flusses die Bewegung des Körpers blockieren können. Des Weiteren ist zu bedenken, dass die Strömungsgeschwindigkeit am Grund des Flusses aufgrund der Reibung geringer ausfällt.

4.1.3 Einfluss der Wassertemperatur auf den menschlichen Körper

Bereits bei Wassertemperaturen unter 25°C kann es zu einer Kälteschockreaktion des Körpers kommen (19). Die Ausprägung des Kälteschocks ist bei 10°C bis 15°C am größten und bleibt dann auch bei sinkender Temperatur gleich stark. Folgen des Kälteschocks können Hyperventilation, gesteigertes Herzzeitvolumen, periphere Vasokonstriktion und Bluthochdruck sein. Dies führt zu einem gesteigerten Sauerstoffbedarf, was die Gefahr des Ertrinkens erhöht. Die Reaktion ist in den ersten 30 Sekunden des Eintauchens am stärksten und lässt während der nächsten 2 bis 3 Minuten ab. Des Weiteren birgt kaltes Wasser die Gefahr der Unterkühlung. Bei Wassertemperaturen um die 0°C besteht bereits nach 30 Minuten akute Lebensgefahr (20). Die größte Gefahr bei der Unterkühlung ist jedoch der Bewusstseinsverlust. Während der Tod bei einer Körpertemperatur von 25 °C eintritt, verliert eine Person

bereits bei einer Körpertemperatur zwischen 33°C bis 30°C das Bewusstsein (19). Bei 35°C Körpertemperatur ist mit einer erheblichen Einschränkung der Handlungsfähigkeit zu rechnen (19, 21).

Tabelle 1 Zeit bis Körpertemperatur auf 35 °C abkühlt in abhängigkeit der Wassertemperatur

Wassertemperatur	Zeit bis die Körpertemperatur auf 35°C abkühlt (21)
20°C	Ca. 1,7h
10°C	Ca. 0,8h
5°C	Ca. 0,5h
0°C	-

4.1.4 Absinken

Für die Berechnung der Sinkgeschwindigkeit einer Person wurde das Modell aus dem Journal of Search and Rescue übernommen (22). Das Modell wurde von einem angloamerikanischen Maßsystem in ein metrisches Einheitensystem umgeändert. Für die Dichte des Wassers wurde, wie in Kapitel 4.2 erläutert, eine Dichte von $\rho=1000\text{kg/m}^3$ eingesetzt. Für die Erdbeschleunigung wurde der Wert von $g=9,81\text{ m/s}^2$ gewählt. Das Modell wurde an das metrische Einheitssystem angepasst. Zur Einfachheit wird der menschliche Körper als Zylinder betrachtet.

Der Radius des Zylinders nach der Einatmung r_1 und der Radius nach der Ausatmung r_2 wird wie folgt berechnet:

4.1.4.1

$$r_1 = \frac{C_1}{2 \cdot \pi}$$

4.1.4.2

$$r_2 = \frac{C_2}{2 \cdot \pi}$$

Wobei C_1 dem Brustumfang nach der Einatmung und C_2 den Brustumfang nach der Ausatmung entspricht. Der Brustumfang nach der Ausatmung entspricht 0.98% des Brustumfangs nach der Einatmung

Das Volumen des Zylinders nach der Einatmung V_1 und der Ausatmung V_2 berechnet sich wie folgt:

4.1.4.3

$$V_1 = h \cdot \pi \cdot r_1^2$$

4.1.4.4

$$V_2 = h \cdot \pi \cdot r_2^2$$

Wobei h die Höhe des Zylinders ist. Die Höhe des Zylinder h entspricht 21% der Höhe der Person.

Setzt man nun 3.1.4.1 in 3.1.4.3 und 3.1.4.2 in 3.1.4.4 ein erhält man:

4.1.4.5

$$V_1 = h \cdot \pi \cdot \frac{C_1^2}{2 \cdot \pi}$$

4.1.4.6

$$V_2 = h \cdot \pi \cdot \frac{(0,98 \cdot C_1)^2}{2 \cdot \pi}$$

Subtrahiert man V_1 und V_2 , erhält man das Volumen V . Es ergibt sich folgende Gleichung:

4.1.4.7

$$V = \frac{h}{4 \cdot \pi} (C_1^2 - (0,98 \cdot C_1)^2)$$

Die Sinkbeschleunigung des Zylinders berechnet sich wie folgt:

4.1.4.8

$$a = \frac{9,81 \frac{m}{s^2} \cdot 1000 \frac{kg}{m^3} \cdot V}{W}$$

Wobei W der Masse der Person entspricht.

Die Gleichung für die Strecke bzw. Tiefe s berechnet sich wie folgt:

4.1.4.9

$$s = \frac{1}{2} \cdot a \cdot t^2$$

Wobei t die Zeit ist.

Formt man nun die Gleichung 4.1.4.9 nach t um und setzt die Gleichungen 4.1.4.8 und 4.1.4.7 ein ergibt sich:

$$t = \sqrt{\frac{2}{9,81 \frac{m}{s^2} \cdot 1000 \frac{kg}{m^3} \cdot \frac{1}{4\pi}}} \cdot \sqrt{\frac{s \cdot W}{h(C_1^2 - (0,98 \cdot C_1)^2)}}$$

Für einen Person mit einem Körpergewicht von 80 kg, einer Größe von 1,8 m und einem Brustumfang nach der Einatmung von 1,1 m. ergeben sich die Absinkzeiten in Tabelle 4.

Tabelle 2: Absinkzeiten in Abhängigkeit der Tiefe für eine Person mit einem Körpergewicht von 80kg, einer Größe von 1,8m

Tiefe in Metern	Zeit in Sekunden
2,5	5,32
5	7,52
7,5	9,21
10	10,64
12,5	11,89
15	13,03
17,5	14,07

4.1.5 Aufschwimmen

Damit ein Körper nach dem Absinken wieder an die Wasseroberfläche auftaucht, müssen genug Fäulnisgase entstehen, die dem Körper den nötigen Auftrieb geben (23). Das Entstehen der Fäulnisgase ist eine Nebenwirkung des Verwesungsprozesses, bei dem sich der Körper aufbläht. Die Geschwindigkeit, in der ein Körper verwest, ist von der Temperatur abhängig. Je niedriger die Temperatur, umso langsamer der Verwesungsprozess und umso länger dauert es, bis der Körper an die Oberfläche kommt. Ein Indikator für die Fortschreitung der Verwesung ist der accumulated degree day (ADD). Der ADD berechnet sich aus der Summe der Produkte der Tage, bei denen der Körper unter Wasser war, und der Temperatur der jeweiligen Tage.

$$ADD = \sum_{i=1}^n (T_i \cdot \frac{h_i}{24})$$

Eine Studie bei der verschiedene Ertrinkungsunfälle verglichen wurden, hat gezeigt, dass es einen Zusammenhang zwischen dem ADD Wert und dem Aufschwimmen von Körpern besteht (23). Es zeigte sich, dass Körper bei einem ADD-Wert von 100 °C-140 °C auftreiben. Die Schlussfolgerung liegt nahe, dass eine Berechnung der Tauchzeit h_{ges} über den ADD-Wert und den Wassertemperaturen berechnet werden kann. Dabei soll hier zur Vereinfachung die Durchschnittstemperatur $T_{durchschnitt}$ genutzt werden. Es ergibt sich folgende Formel:

$$h_{ges} = \frac{ADD}{T_{durchschnitt}}$$

Anhand der Formel¹ wurden für unterschiedliche Wassertemperaturen, Zeiten bis zum Auftauchen des Körpers berechnet.

Tabelle 3 Zusammenhang zwischen Wassertemperatur und ADD

Wassertemperatur	Tauchzeit für ADD 100°C in Tagen	Tauchzeit für ADD 140°C in Tagen
5	20	28
10	10	14
15	6,6	9,3
20	5	7
25	4	5,6

4.2 Auswertung der Simulationen

Ab dem 01.06.2022 war das Abrufen des Webfeatures OpCIS mit langen Ladezeiten verbunden. Dadurch war es ab diesem Zeitpunkt nicht mehr möglich Simulationen durchzuführen. Grund hierfür war, dass der Server in diesem Zeitraum durch umfangreiche Umstrukturierungen von einer weiteren Applikation genutzt wurde. Da das Modell, wie es in Kapitel 3.2.6 vorgestellt wurde, erst am 30.05.2022 fertiggestellt wurde, ist die Anzahl der Simulationen mit diesem Modell auf eine Simulation begrenzt. Aus diesem Grund werden hier auch Simulationen vorgestellt, bei denen das Modell noch nicht den Widerstandbeiwert berücksichtigt. Der Widerstandbeiwert ist lediglich bei der Simulation in Övelgönne eingebaut.

Die Auswertung wird nachfolgend anhand von Simulationen an sechs unterschiedlichen Standorten durchgeführt. Dabei werden zur Auswertung die Distanzen zwischen dem Startpunkt der Simulation und der Punkte in Abständen von 30 Sekunden, 3 Minuten, 5 Minuten, 10 Minuten, 15 Minuten, 20 Minuten, 25 Minuten und 30 Minuten nach Start der Simulation tabellarisch dargestellt. Des Weiteren wird der Streckenverlauf visualisiert. Hierfür wurden die auf der Webkarte ausgegebenen Punkte verbunden und die Punkte durch Pfeile ersetzt. Die Pfeile stellen somit die Richtung dar, in der die Person bei der Simulation getrieben wird, sowie die Aufenthaltspunkte, an denen die Person nach 5 Minuten, 10 Minuten, etc. aufhält. Es wird angegeben, ob am Standort St. Pauli Landungsbrücken, zum Zeitpunkt der Simulation, das Wasser ablaufend oder auflaufend war. Zudem wird die Zeit bis zum nächsten Scheitelpunkt am Standort St. Pauli angegeben. Weitere Simulationen sind im Anhang zu finden.

Bei der Standortwahl wurde darauf geachtet, dass sich die Standorte sowohl an den Haupt- wie in den Nebenströmen befanden. Des Weiteren wurden mit dem Elbstrand in Övelgönne und dem Elbpark Entenwerder zwei beliebte Badeorte ausgewählt. So konnten Badeunfälle simuliert werden. Zwei Standorte befinden sich zudem auf den

Brücken Köhlbrandbrücke und Mahatma-Ghandi Brücke. Diese Brücken wurden gewählt, um einen Sprung, bzw. Fall von einer Brücke zu simulieren.

Am Standort Mahatma-Ghandi Brücke wurden 3 Durchläufe mit einem Abstand von einer Stunde durchgeführt. Dabei war der erste Durchlauf 4 Stunden und 12 Minuten vor Niedrigwasser an den St.-Pauli-Landungsbrücken. Es kam hierbei lediglich zu einer Bewegung für Fall 1. Die Distanz im Durchlauf 1 für Fall 1 betrug 83,59 Meter und die Person wurde in der Simulation Richtung Landungsbrücken getrieben. Im zweiten Durchlauf kam es für Fall 1 zu einer größeren Distanz von 92,6 Metern. Anders als im Durchlauf 1 kam es im Durchlauf 2 im Fall 3, zu einer Bewegung. Jedoch wurde die Person erst 20 Minuten nach Simulationsbeginn bewegt. In beiden Fällen wurde die Person Richtung Landungsbrücken getrieben. Im Durchlauf 3 kam es zu wesentlich kürzeren Strecken. Dadurch kam es im Fall 1 lediglich zu einer Distanz von 33,4 Metern. Auch hier kam es erst im späten Verlauf der Simulation zu einer Bewegung für Fall 3. Es zeigte sich, dass es bei den Fällen 4 und 5, die an Land waren, nicht zu Bewegungen kam. Ebenso verhielt es sich mit Fall 2, wobei sich Fall 2 im Wasser befand.

Tabelle 4 Simulation Mahatma-Gandhi-Brücke Durchlauf 1

Datum:	Uhrzeit Ende	Ort			Auflauf oder ablaufend	Zeitpunkt bis Scheitelpunkt
23.5.2022	13:50	Mahatma-Gandhi-Brücke			Ablaufend	4:12
Zeit	Distanz Fall 1	Distanz Fall 2	Distanz Fall 3	Distanz Fall 4	Distanz Fall 5	
30 sec	1,54321	0	0	0	0	
3 min	4,783951	0	0	0	0	
5 min	6,481481	0	0	0	0	
10 min	21,86214	0	0	0	0	
15 min	37,29424	0	0	0	0	
20 min	52,6749	0	0	0	0	
25 min	68,107	0	0	0	0	
30 min	83,59053	0	0	0	0	

Tabelle 5 Simulation Mahatma-Gandhi-Brücke Durchlauf 2

Datum:	Uhrzeit Ende	Ort		Auflauf oder ablaufend	Zeitpunkt bis Scheitelpunkt	
23.5.2022	14:50	Mahatma-Gandhi- Brücke		Ablaufend	3:12	
Zeit	Distanz Fall 1	Distanz Fall 2	Distanz Fall 3	Distanz Fall 4	Distanz Fall 5	
30 sec	1,54321	0	0	0	0	
3 min	9,259259	0	0	0	0	
5 min	15,4321	0	0	0	0	
10 min	30,8642	0	0	0	0	
15 min	46,2963	0	0	0	0	
20 min	61,67695	0	0,41152	0	0	
25 min	77,10905	0	3,49794	0	0	
30 min	92,59259	0	4,06378	0	0	

Tabelle 6 Simulation Mahatma-Gandhi-Brücke Durchlauf 3

Datum:	Uhrzeit Ende	Ort		Auflauf oder ablaufend	Zeitpunkt bis Scheitelpunkt
23.5.2022	15:50	Mahatma-Gandhi- Brücke		Ablaufend	2:12
Zeit	Distanz Fall 1	Distanz Fall 2	Distanz Fall 3	Distanz Fall 4	Distanz Fall 5
30 sec	0,205761	0	0	0	0
3 min	0,874486	0	0	0	0
5 min	1,49177	0	0	0	0
10 min	2,572016	0	0	0	0
15 min	2,572016	0	0	0	0
20 min	2,572016	0	0	0	0
25 min	17,90123	0	2,98354	0	0
30 min	33,38477	0	3,90946	0	0



Abbildung 30 Simulation Mahatma-Gandhi-Brücke

Tabelle 7 Simulation Köhlbrandbrücke

Datum:	Uhrzeit Ende	Ort		Auflaufend oder Ablaufend	Zeitpunkt bis Scheitelpunkt
13.5.2022	16:10	Köhlbrandbrücke		Ablaufend	6:20
Zeit	Distanz Fall 1	Distanz Fall 2	Distanz Fall 3	Distanz Fall 4	Distanz Fall 5
30 sec	6,17284	6,378601	4,783951 0	4,783951	4,783951
3 min	33,38477	33,59053	24,12551	24,17695	27,77778
5 min	51,90329	51,95473	36,7284	36,52263	46,2963
10 min	92,38683	118,7757	67,84979	67,38683	82,15021
15 min	128,8066	118,7757	88,78601	88,47737	103,035
20 min	151,4403	141,6152	104,2181	95,67901	118,4671
25 min	155,5041	145,5761	108,179	96,5535	122,3765
30 min	166,358	156,3786	117,9527	110,3909	133,0761

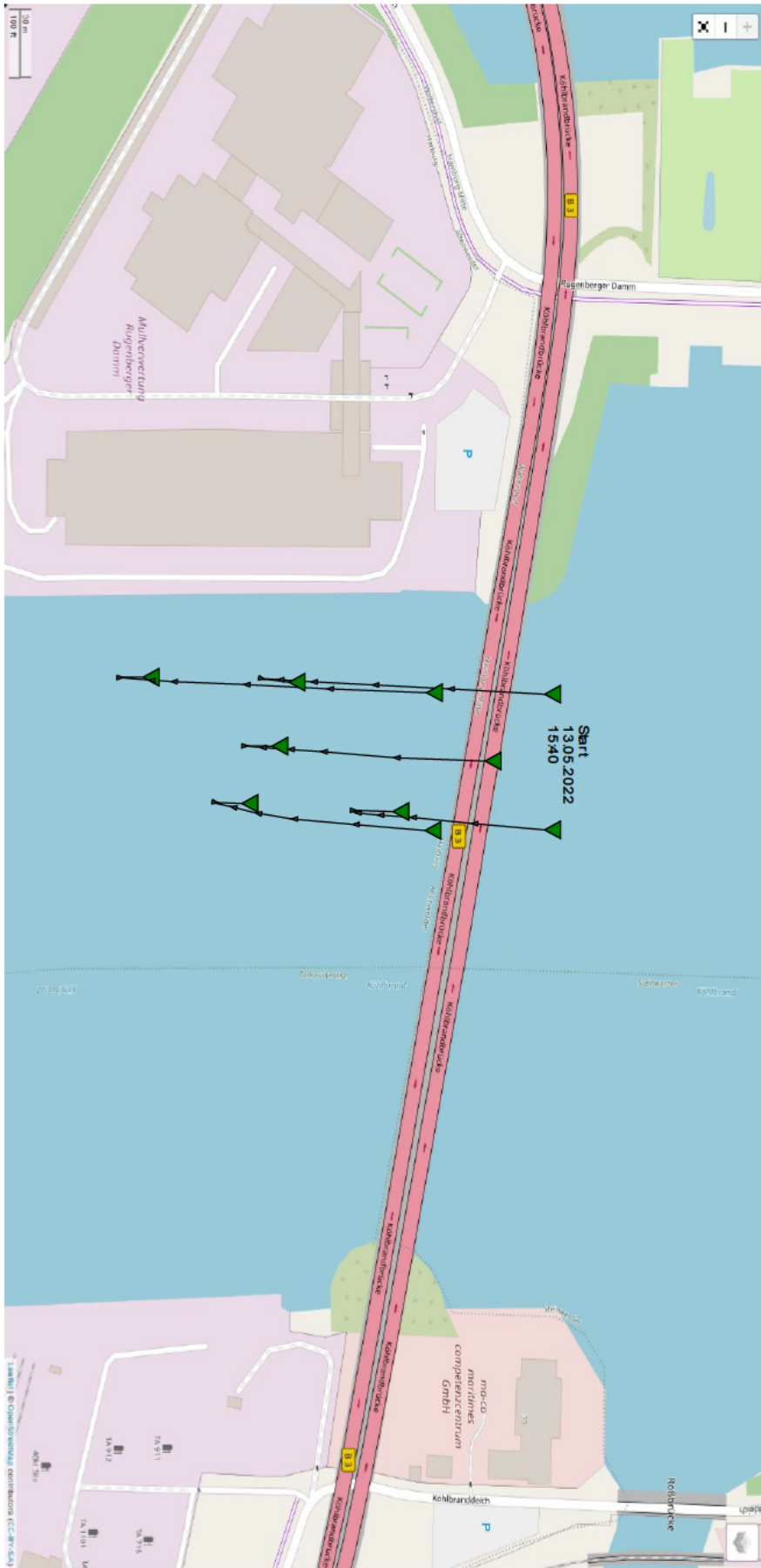


Abbildung 31 Simulation Köhlbrandbrücke

Tabelle 8 Simulation Veddeler Kanal

Datum:	Uhrzeit Ende	Ort		Auflaufend oder ablaufend	Zeitpunkt bis Scheitelpunkt
16.5.2022	11:30	Veddeler Kanal		Auflaufend	4:00
Zeit	Distanz Fall 1	Distanz Fall 2	Distanz Fall 3	Distanz Fall 4	Distanz Fall 5
30 sec	0	1,59465	2,366255 0	2,211934	1,59465
3 min	0	9,41358	9,3107	22,83951	9,259259
5 min	0	15,58642	9,36214	29,01235	15,4321
10 min	0	46,2963	9,36214	44,29012	30,76132
15 min	0	46,2963	9,36214	53,60082	36,7284
20 min	0	56,99588	9,36214	53,60082	36,7284
25 min	0	56,99588	9,36214	53,60082	41,25514
30 min	0	62,65432	9,36214	53,60082	56,68724



Abbildung 32 Simulation Veddeler Kanal

Tabelle 9 Simulation Elbpark Entenwerder

Datum:	Uhrzeit Ende	Ort	Auflaufend oder ablaufend	Zeitpunkt bis Scheitelpunkt	
18.5.2022	13:55	Elbpark Entenwerder	Ablaufend	0:07	
Zeit	Distanz Fall 1	Distanz Fall 2	Distanz Fall 3	Distanz Fall 4	Distanz Fall 5
30 sec	15,79218	38,42593	11,16255	0	10,13374
3 min	91,82099	84,41358	73,35391	0	98,81687
5 min	157,0988	88,78601	136,1626	0	130,5041
10 min	315,4835	88,78601	251,3374	0	141,358
15 min	437,1399	88,78601	353,7551	0	144,1872
20 min	539,5062	88,78601	461,9342	0	144,1872
25 min	675,4115	88,78601	549,4856	0	144,1872
30 min	787,4486	88,78601	656,2243	0	144,1872

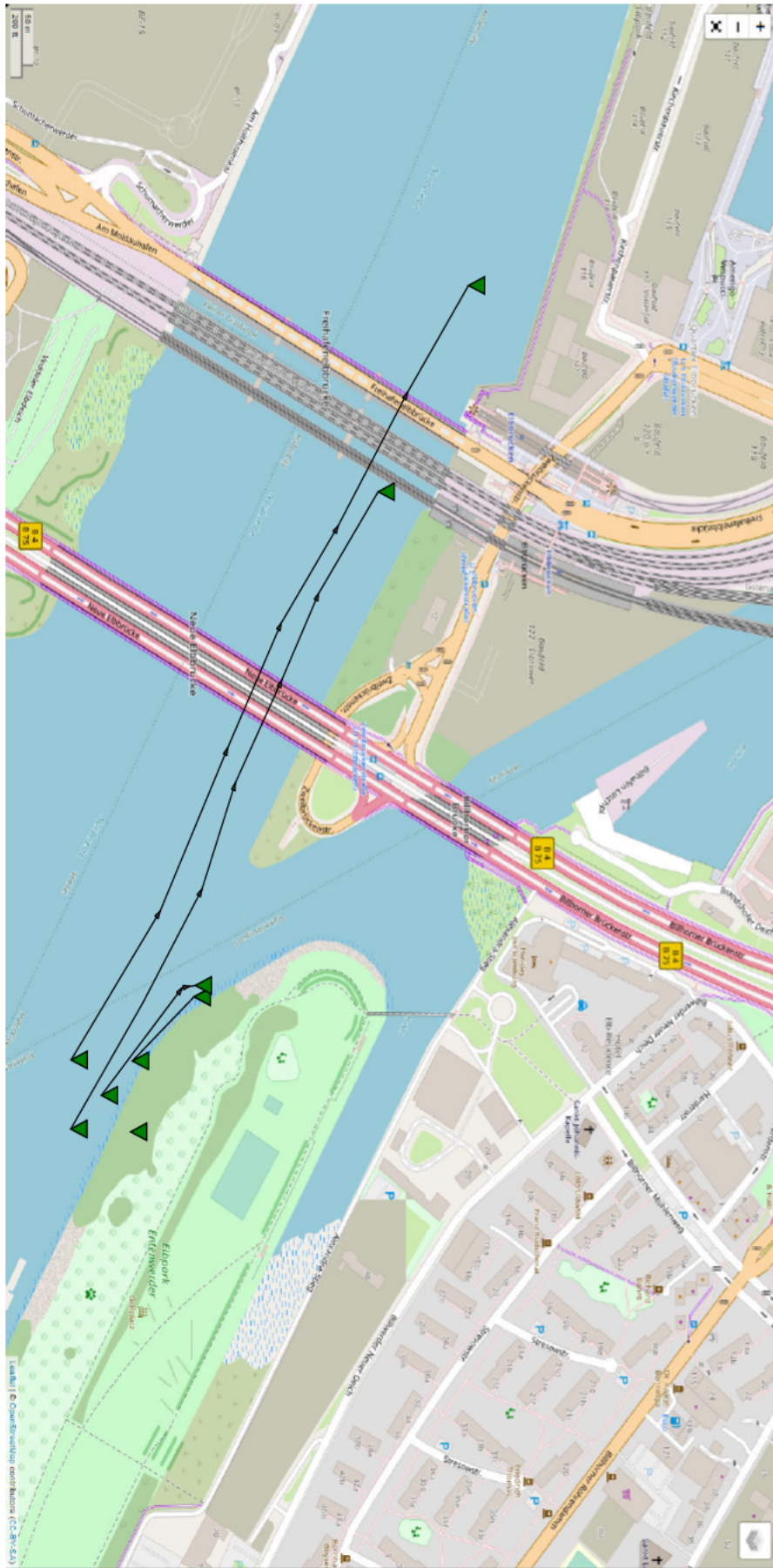


Abbildung 33 Simulation Elbpark Entenwerder

Tabelle 10 Simulation Flugplatz Hamburg Finkenwerder

Datum:	Uhrzeit Ende	Ort		Auflaufend oder ablaufend	Zeitpunkt bis Scheitelpunkt
30.5.2022	10:20	Flugplatz Hamburg Finkenwerder		Auflaufend	2:17
Zeit	Distanz Fall 1	Distanz Fall 2	Distanz Fall 3	Distanz Fall 4	Distanz Fall 5
30 sec	41,81039	42,06931	38,3039	40,95376	40,30799
3 min	224,5889	230,3444	211,5896	225,1111	221,4515
5 min	354,6193	371,2337	345,9777	368,3543	359,2986
10 min	692,7328	723,5863	682,7974	722,1016	705,0518
15 min	991,5624	1055,051	1000,792	1052,726	1033,269
20 min	1271,181	1372,666	1308,067	1368,264	1347,144
25 min	1493,481	1668,024	1582,987	1664,023	1631,192
30 min	1683,221	1953,931	1853,824	1953,931	1908,755

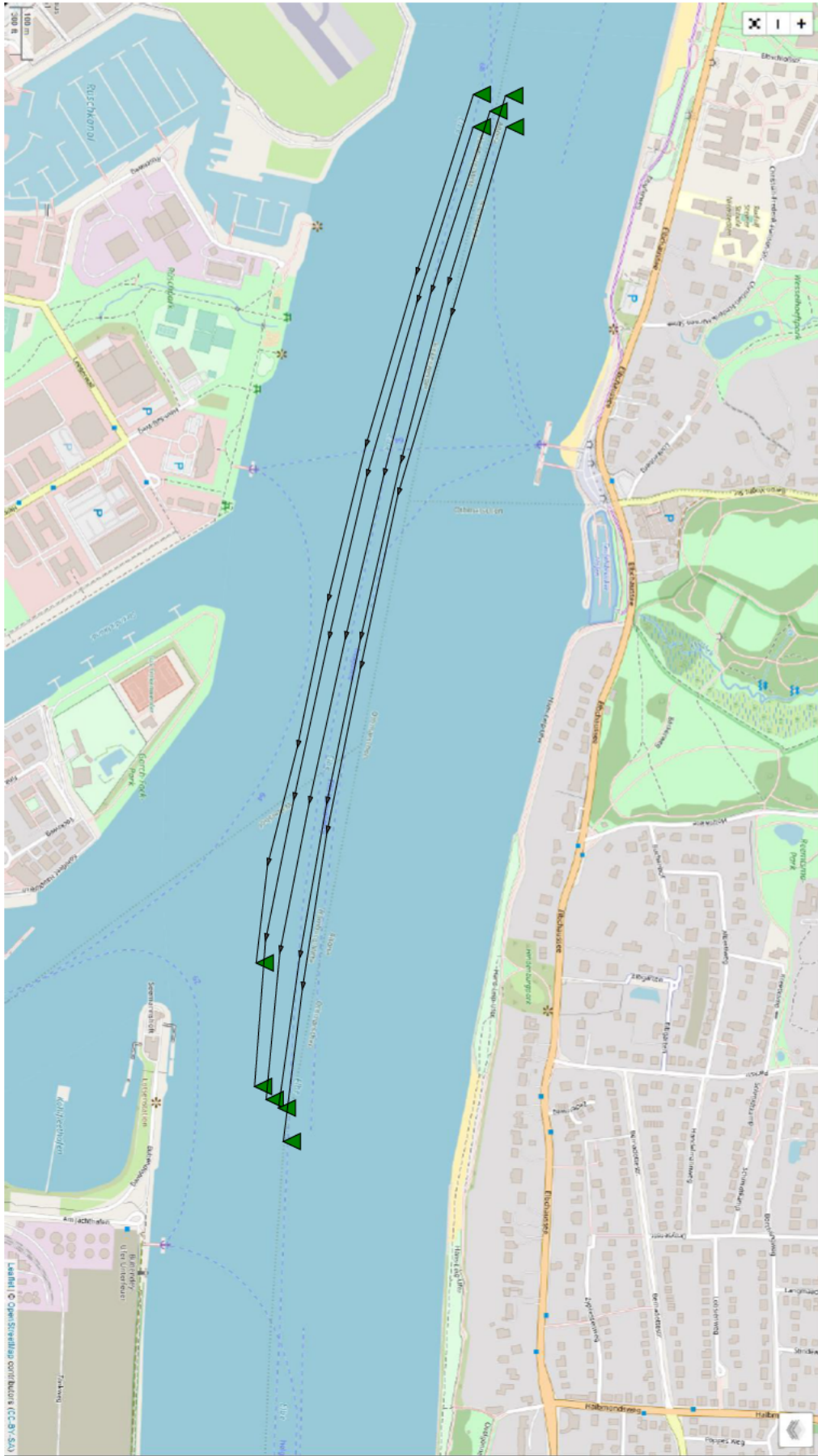


Abbildung 34 Simulation Flughafen Hamburg Finkenwerder

Tabelle 11 Simulation Övelgönne

Datum:	Uhrzeit Ende	Ort		Auflaufend oder ablaufend	Zeitpunkt bis Scheitelpunkt
31.5.2022	13:30	Övelgönne		Auflaufend	2:17
Zeit	Distanz Fall 1	Distanz Fall 2	Distanz Fall 3	Distanz Fall 4	Distanz Fall 5
30 sec	0,073588	0	0	0	0
3 min	0,441529	0	0,03146	0	0
5 min	6,976166	0	27,91091	0	0
10 min	39,46292	0	251,3374	0	0
15 min	146,2419	0	353,7551	0	0
20 min	283,8372	0	461,9342	0	0
25 min	283,8372	0	549,4856	0	0
30 min	283,837234224 964	0	1004,541956	0	0

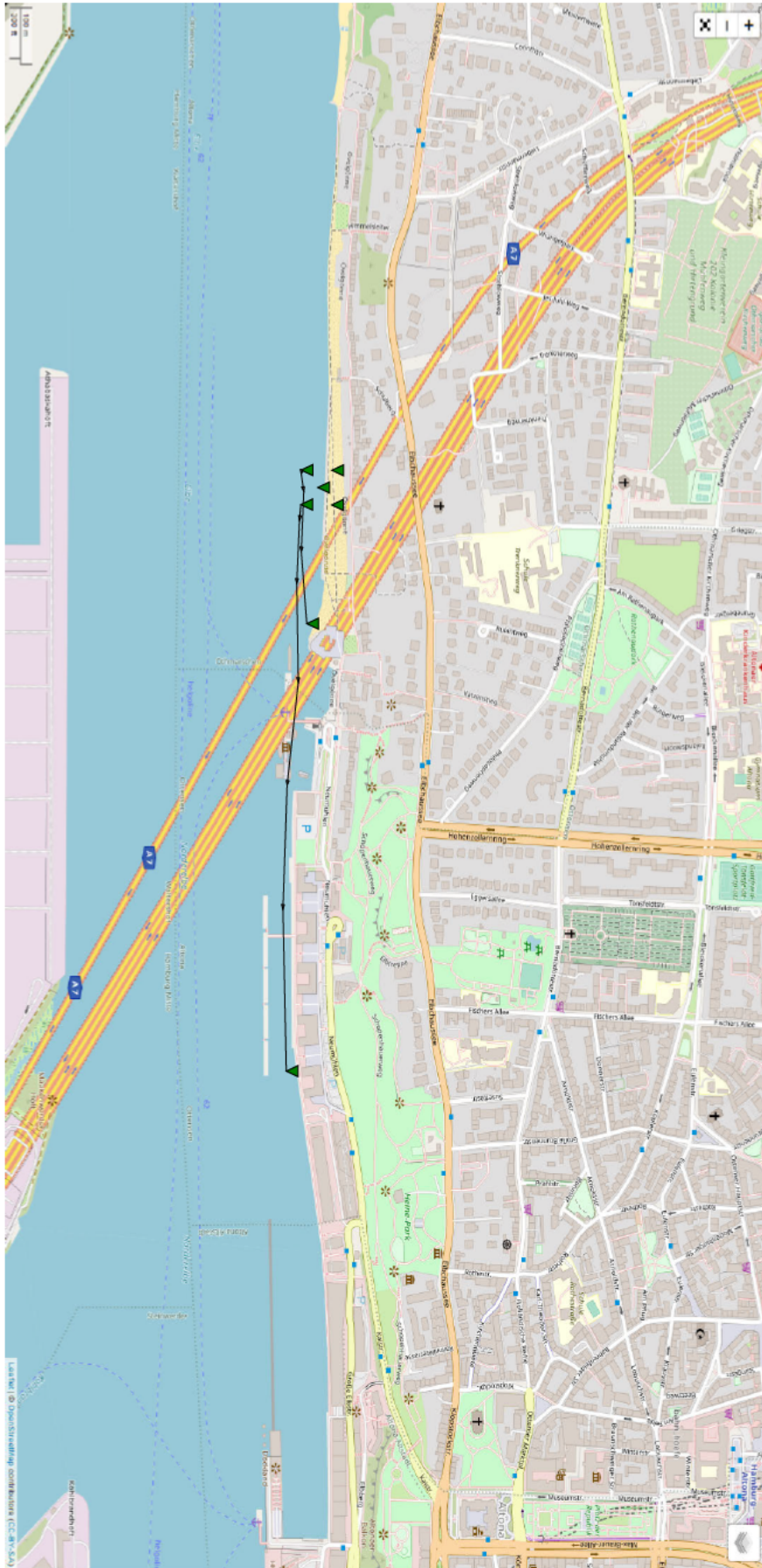


Abbildung 35 Simulation Övelgönne

An Punkten, für die aufgrund ihres Standortes (Fall 2 und Fall 4 Övelgönne, Fall 2 und Fall 4 Mahatma-Gandhi Brücke) außerhalb des Wassers keine Strömungsdaten zugrunde lagen, wurde keine Distanz bzw. keine Bewegung berechnet. Zu beobachten war zudem, dass es bei Punkten in Ufernähe vermehrt zu kürzeren Distanzen kam. Grund hierfür waren geringe Strömungsgeschwindigkeiten (Fall 5 Övelgönne, Fall 5 Durchlauf 2 und 3) oder die Anlandung durch die Strömungsrichtung in Richtung Ufer (Fall 1 Övelgönne, Fall 2 und 5 Entenwender). Selbst bei Punkten mit weiter Entfernung zum Ufer (Flugplatz Hamburg Finkenwerder) waren an den Punkten, die näher am Ufer waren, geringere Strömungsgeschwindigkeiten zu beobachten. Auffällig ist zudem, dass es zwischen den verschiedenen Standorten zu starken Abweichungen in den gemessenen Distanzen kam. So wurden am Standort Flugplatz Hamburg Finkenwerder Distanzen von bis zu 1953 m berechnet. Während es am Standort Mahatma-Ghandi Brücke lediglich zu einer maximalen Distanz von 92,6 m kam. Allgemein waren die gemessenen Distanzen in den Hauptströmen der Elbe größer als die in den Nebenarmen. Am Standort Mahatma-Gandhi Brücke konnte bis zum zweiten Durchlauf beobachtet werden, dass in den Ersten beiden Durchläufen wesentlich längere Distanzen erreicht wurden als im dritten Durchlauf, der nur 2 Stunden und 12 Minuten vor Niedrigwasser beendet wurde. Im Standort Köhlbrandbrücke kam es während der Simulation zum Wechsel zwischen Ebbe und Flut. Hier konnte beobachtet werden, dass die Distanz zunehmend abnahm bis es zum Richtungswechsel aufgrund der Gezeitenänderung kam.

5 Diskussion

5.1 Verweildauer eines menschlichen Körpers auf und unter Wasser

Aus den in Kapitel 4.1 vorgestellten Informationen lassen sich Schlussfolgerungen auf die Zeit machen, während der, eine Person von der Strömung bewegt wird. Grund hierfür ist, dass eine Person, wie in Kapitel 4.1.2 gezeigt, beim Aufsetzen auf den Grund des Gewässers nicht mehr von der Strömung bewegt wird. Somit wird eine Person nur dann von der Strömung bewegt, wenn sie sich an der Wasseroberfläche befindet oder absinkt.

Wie lange eine Person maximal an der Wasseroberfläche bleiben kann, lässt sich anhand der Wassertemperatur abschätzen. Grund hierfür ist, dass Menschen im Verlauf der Unterkühlung das Bewusstsein verlieren. Da die meisten Menschen aufgrund der Dichte des menschlichen Körpers spätestens bei der Ausatmung absinken, liegt die Schlussfolgerung nahe, dass bewusstlose Personen oder Personen die nicht fähig sind zu schwimmen, absinken werden.

Wie lange eine Person bei Bewusstsein bleibt, hängt von der Ausprägung der Kälteschockreaktion und der Körpertemperatur ab. Je niedriger die Wassertemperatur ist, umso schneller fällt die Körpertemperatur und desto früher verliert eine Person das Bewusstsein. Besonders die Kälteschockreaktion birgt große Gefahren. Da hier ein Mensch bei Temperaturen unter 15 °C in den ersten 3 Minuten die Schwimmfähigkeit verlieren kann. Zwar kommt es in der Elbe, in der Mehrheit der Tage im Jahr zu einer Wassertemperatur unter 15°C, jedoch ist die Wahrscheinlichkeit einer Kältereaktion, die mit dem Verlust der Schwimmfähigkeit verbunden ist, gering.

Auch die Zeit, die ein Körper benötigt, um auf den Grund zu sinken, lässt sich bestimmen. So benötigt ein erwachsener Mann selbst bei einer Wassertiefe von 16,7 m nur wenige Sekunden bis er zum Grund absinkt. Erst wenn der Körper aufgrund von Fäulnisprozessen auftreibt, wird er wieder von der Strömung bewegt.

Es kann also davon ausgegangen werden, dass sobald eine Person ihre Schwimmfähigkeit verliert, absinkt und nur wenige Meter weiter getrieben wird.

In Abbildung 36 werden unterschiedliche Zustände betrachtet, in denen sich ein menschlicher Körper im Wasser befinden kann. Den Zuständen 1 bis 3 aus der nachfolgenden Abbildung werden verschiedene Zeiten zugeordnet. Im Zustand 1 ist die Person noch bei Bewusstsein und in der Lage sich über Wasser zu halten. Im Zustand 2 hat die Person bereits die Fähigkeit verloren sich aktiv über Wasser zu halten. Der Sinkprozess beginnt. Im Zustand 3 hat der Körper die Sohle des Gewässers erreicht. Im Zustand 4 steigt der Körper durch Bildung von Fäulnisgasen auf und gelangt wieder an die Wasseroberfläche.

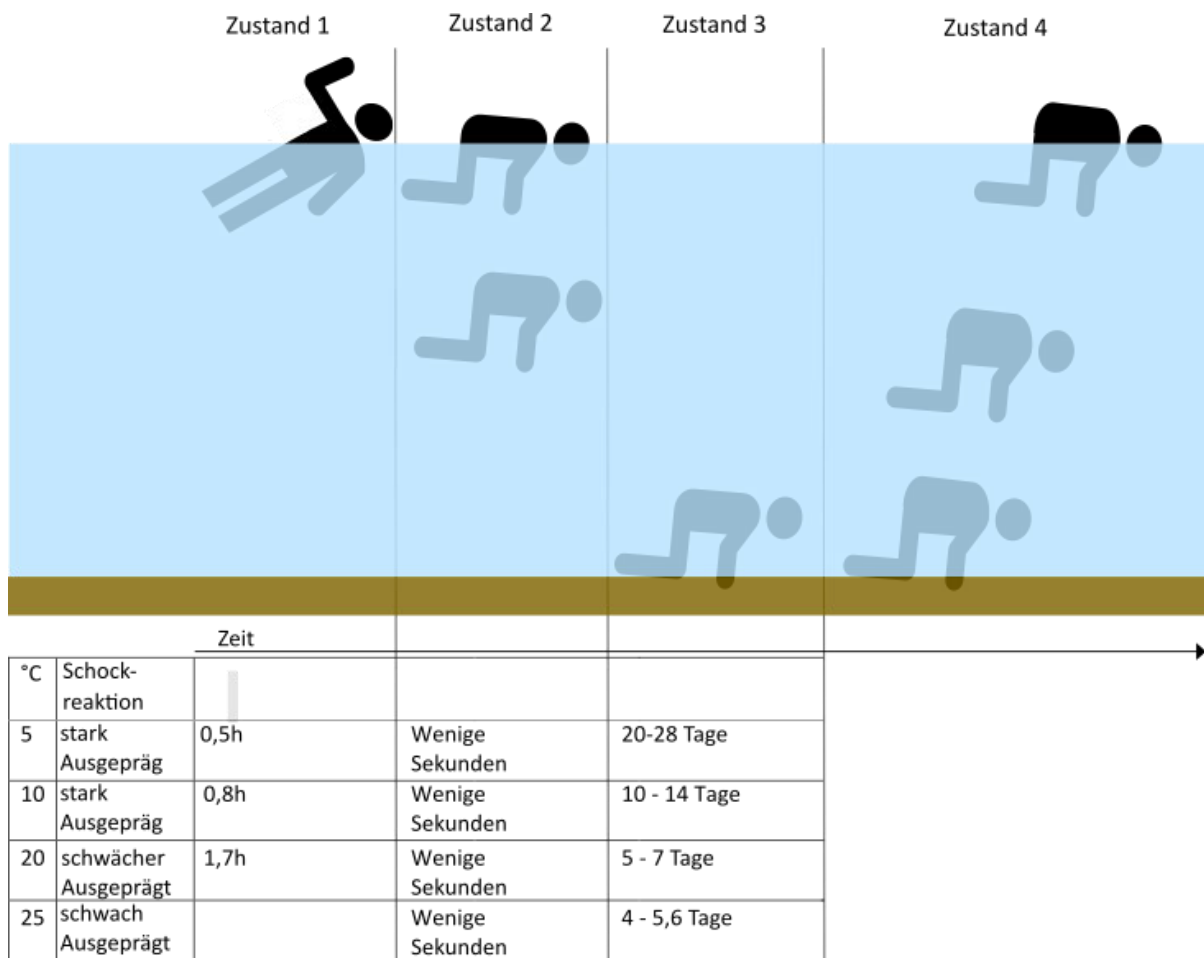


Abbildung 36 Verweildauer in Abhängigkeit der Zeit

5.1.1 Auswertung der Modellierung

Im Rahmen dieser Arbeit wurde ein Programm in C# erstellt, welches erste Prognosen über das Treibverhalten ermöglicht. So konnte deutlich der Einfluss der Strömungsgeschwindigkeit und Strömungsrichtung gezeigt werden. Eine Simulation des Treibverhaltens einer Person in Echtzeit war anhand dieses Modells möglich.

Aufgrund der geringen Anzahl an Simulationen ist es schwierig, allgemeine Aussagen über das Treibverhalten zu machen. Allerdings soll hier der Versuch unternommen werden, Schlüsse anhand der bestehenden Daten zu ziehen. So kann man, an den Daten sehen, dass die Bestimmung des genauen Ortes an dem die Person das letzte Mal im Wasser gesichtet wurde, einen erheblichen Einfluss auf die Distanz und den Streckenverlauf haben kann.

Des Weiteren zeigt sich, dass anhand der Zeit bis zum nächsten Scheitelpunkt, eine Abschätzung bezüglich der Strecke und Distanz gemacht werden kann. So ist mit geringen Geschwindigkeiten, wenige Minuten vor und nach dem Gezeitenwechsel, zu rechnen. Der Zeitpunkt für den Gezeitenwechsel ist für jeden Punkt in der Elbe unterschiedlich.

In den Hauptströmen kann bei auflaufendem Wasser mit einer Distanz von bis zu 230 m in den ersten 3 Minuten gerechnet werden. Nach 30 Minuten könnte eine Person über 1900 m weit getrieben worden sein. Es kann davon ausgegangen werden, dass sich bei ablaufendem Wasser ähnliche Werte ergeben.

Aus den Simulationen geht zudem hervor, dass die Strömungsgeschwindigkeiten stark vom Ort abhängig sind. In den Hauptströmen der Elbe ist mit deutlich größeren Geschwindigkeiten zu rechnen als in den Nebenströmen.

Die Nähe zum Ufer scheint einen Einfluss auf das Treibverhalten zu haben. So kam es in Ufernähe zu kürzeren Strecken, da hier die Strömungsgeschwindigkeit geringer ist und es zur Anlandung der Person kommen kann.

6 Fazit und Ausblick

Ziel dieser Arbeit war es, ein Modell zu erstellen, das das Treibverhalten von vermissten Personen im Hamburger Hafen darstellen kann. Hierzu wurde gezeigt wie ein menschlicher Körper über und unter Wasser von der Strömung bewegt wird. Dabei ergab sich, dass Personen, die nicht fähig sind zu Schwimmen, mit einer hohen Wahrscheinlichkeit nur wenige Meter weit getrieben werden. Zudem wurde die maximale Verweildauer des menschlichen Körpers an der Wasseroberfläche und unter Wasser bestimmt. Für die Berechnung des Drifts einer Person, durch die Strömung, wurde ein Modell erstellt, das anhand aktueller Strömungsdaten die Bewegung einer Person, durch die Strömung simuliert. Es wurden Ergebnisse dieses Modells zusammengestellt und analysiert.

In seiner jetzigen Form simuliert das Modell bereits die Bewegung einer Person anhand aktueller Strömungsdaten. Die begrenzte Anzahl an durchgeführten Simulationen macht es schwierig, allgemein zutreffende Aussagen zu machen. Grund hierfür waren lange Ladezeiten aufgrund von Wartungsarbeiten am Server. Jedoch konnte gezeigt werden, welchen großen Einfluss der Verlauf der Tide, der Standorte, sowie die Nähe zum Ufer auf das Treibverhalten haben kann. Es kann damit gerechnet werden, dass nach den Wartungsarbeiten die Ladezeit wieder kurz genug sein wird, um weitere Simulationen durchzuführen. Mit der Auswertung von weiteren Simulationen könnten allgemeine Aussagen getroffen werden, die einer besseren Eingrenzung des Suchsektors dienen könnten. Es wäre vorstellbar, dass für unterschiedliche Tiden maximale Distanzen für Haupt- und Nebenströme angegeben werden könnten.

Im Rahmen dieser Arbeit konnte das Modell nicht experimentell überprüft werden. Für eine Überprüfung der Praxistauglichkeit wären Versuche mit Dummies oder Probanden nötig. Auf diese Weise könnten die Simulationen experimentell nachgestellt werden und gegebenenfalls das Modell angepasst werden.

Eine weitere Limitation des Modells ist der Widerstandbeiwert, da dieser von individuellen Faktoren der gesuchten Person abhängt und sich von Person zu Person unterscheidet. In Zukunft könnten hierfür weitere Widerstandbeiwerte für unterschiedliche Bekleidungen und Körpertypen bestimmt werden. Anhand der Beschreibung der meldenden Person über Geschlecht, Körpergröße, Kleidung etc., könnten so passende Widerstandbeiwerte gewählt werden, die als drop down Menu in das Programm integriert werden könnten. Auch die Wassertemperatur könnte in das Modell mit eingebaut werden und somit wären Aussagen über die maximale Drift Zeit möglich.

Die Abhängigkeit einer schnellen Ladezeit des Webfeatures hat sich ebenfalls als Problem dargestellt. Wie sich im Verlaufe der Arbeit zeigte, ist eine Simulation bei zu langen Ladezeiten nicht möglich. Zur Optimierung könnte man, statt online auf die Daten zu zugreifen, das Modell auf direkte Weise mit den Daten arbeiten lassen. Es wäre auch vorstellbar die Verteilung der Polygone auf dem berechnenden Gerät zu speichern. So müsste nur bei Erreichen eines neuen Polygons auf die online Strömungsdaten zurückgegriffen werden, was die Anzahl an Abrufen der Webseite verkürzen würde.

Auch die Möglichkeit von Vorhersagen wäre mit dem Modell möglich. Hierfür wäre es notwendig, Strömungsdaten aus der Vergangenheit und Vorhersagen der Strömungsdaten für die Zukunft zu verwenden. Die Grundstruktur des Modells könnte aber erhalten bleiben. Es könnten beispielweise alle fünf Minuten Strömungsdaten, die 30 Minuten in der Zukunft liegen, verwendet und für eine Stunde gespeichert werden. Das Programm könnte dann auf diese Daten zurückgreifen und simulieren, wie eine Person in 30 Minuten fortgetrieben sein könnte. Anhand der Simulation würde man den Suchsektor genauer eingrenzen können.

Das in dieser Arbeit entwickelte Modell und die Recherchearbeit liefert bereits vielversprechende Ergebnisse, die einen Einblick in das Treibverhalten von vermissten Personen im Hamburger Hafen geben. Während ein Großteil der Arbeit darin bestand, das Modell zu entwickeln und eine passende Umsetzung zu finden, ist das Potential des Modells noch nicht ausgeschöpft. So bildet die Arbeit eine Grundlage für nachfolgende Arbeiten und Anwendungen, die die Personensuche im Hamburger Hafen erleichtern können und zur Rettung von Menschenleben in Zukunft beitragen kann.

7 Anhang

Der Anhang dieser Arbeit befindet sich auf einem USB-Stick, welcher am Ende angeheftet wurde. Dort sind die erstellten Excel Tabellen der vorgestellten Simulationen, sowie die Screenshots der zugehörigen Visualisierung auf Geoplaner zu finden. Des Weiteren ist das auf C# geschriebene Programm in der vorgestellten, aktuellen Form auf dem USB-Stick gespeichert. Es ist zudem die Arbeit in einer digitalen Version zu finden. Im Ordner Screenshots sind die Bildschirmaufnahmen der Simulation zu finden und im Ordner Excel die dazugehörigen Tabellen mit Koordinaten, Geschwindigkeiten und Streckenangaben zu finden. Die Dateien wurden nach dem Datum und der Uhrzeit für das Ende der Simulation benannt. Im Ordner C# ist das Modell zu finden.

Nachfolgend ist eine Übersicht über die Ordnerstruktur des Anhangs aufgeführt:

- Treibverhalten von vermissten Personen am Beispiel des Hamburger Hafens
- Screenshots
 - 13.05.2022_1610
 - 16.05.2022_1130
 - 16.05.2022_1652
 - 18.05.2022_1355
 - 19.05.2022_1717
 - 23.05.2022_1355
 - 23.05.2022_1455
 - 23.05.2022_1555
 - 23.05.2022_1655
 - 30.05.2022_1020
 - 31.05.2022_1330
- Excel
 - 13.05.2022_1610
 - 16.05.2022_1130
 - 16.05.2022_1652
 - 18.05.2022_1355
 - 19.05.2022_1717
 - 23.05.2022_1355
 - 23.05.2022_1455
 - 23.05.2022_1555
 - 23.05.2022_1655
 - 30.05.2022_1020
 - 31.05.2022_1330

- C#
 - ConsoleApp

8 Literaturverzeichnis

1. Einsatz: Personensuche vom 02.08.2015; 2022 [Stand: 30.05.2022]. Verfügbar unter: <https://wedel.dlrg.de/retten/einsatztagebuch/personensuche-7362-e/#gallery-4>.
2. Lück T, Kvasnicka V, Lück A. Unterwasserdrohnen in der Wasserrettung. Unfallchirurg 2021; 124(12):977–83. Verfügbar unter: <https://link.springer.com/article/10.1007/s00113-021-01097-1>.
3. Christoph Slaby. Einsatzstatik der Feuwehr Wasserrettung; 2011 [Stand: 31.05.2022]. Verfügbar unter: <https://www.lfs-bw.de/fileadmin/LFS-BW/themen/einsatzdienst/sonderlagen/dokumente/Wasserrettung.pdf>.
4. Fahrrinnenanpassung; 2022 [Stand: 07.06.2022]. Verfügbar unter: <https://www.hamburg-port-authority.de/de/hpa-360/bauprojekte/fahrrinnenanpassung>.
5. Stephan Kohfahl. Persönliche Schutzausrüstung für Einsatzkräfte der Feuerwehr im Wasserrettungseinsatz bei niedrigen Wassertemperaturen [Bachelorarbeit]. Hamburg: Hochschule für Angewandtwissenschaften Hamburg.
6. Michel Gérard. Strömungs- und Durchflussmessungen im Bereich der Tideelbe seit Anfang der 1960er Jahre: Ergebnisse aus Untersuchungen der Wasser- und Schifffahrtsverwaltung im Bereich Elbe-km 638,9 und 689,1 [Masterarbeit]. Braunschweig/Wolfenbüttel: Ostfalia Hochschule für angewandte Wissenschafte; 2012.
7. Böswirth L, Bschorer S. Technische Strömungslehre: Lehr- und Übungsbuch. 9., überarbeitete Auflage. Wiesbaden: Vieweg+Teubner Verlag; 2012. (SpringerLink Bücher).
8. Harten U. Physik: Eine Einführung für Ingenieure und Naturwissenschaftler. 5. Aufl. Berlin, Heidelberg: Springer; 2012. (Springer-Lehrbuch).
9. Spektrum. UTM: Lexikon der Geographie. Heidelberg: Spektrum Akademischer Verlag. Verfügbar unter: <https://www.spektrum.de/lexikon/geographie/utm/8499>.
10. John P. Snyder. Map Projection: A Working Manual: U.S. Government Printing Office; 1987. (Geological Survey professional paper; 1395.).
11. Dr.-Ing. Oliver Stoschek, Simone McCurdy, Dr. Arne, Thomas Strotmann. Echtzeitströmungsmodell für den Hamburger Hafen. Schiff & Hafen 2016; (4):26–8 [Stand: 05.05.2022]. Verfügbar unter: https://www.schiffundhafen.de/fileadmin/user_upload/maritim40/Fachbeitraege_pdf/Echtzeitstroemungsmodell.pdf.
12. Kolmogorov SV, Duplishcheva OA. Active drag, useful mechanical power output and hydrodynamic force coefficient in different swimming strokes at maximal velocity.

- Journal of Biomechanics 1992; 25(3):311–8. Verfügbar unter:
<https://www.sciencedirect.com/science/article/pii/002192909290028y>.
13. Hartwig Harm. Umrechnung zwischen geographischen Koordinaten (Länge und Breite) und dem UTM-Format [Stand: 19.05.2022]. Verfügbar unter:
<https://docplayer.org/24003599-Umrechnung-zwischen-geographischen-koordinaten-laenge-und-breite-und-dem-utm-format.html>.
 14. Llana-Belloch A., Lucas-Cueva A.G., Perez-Soriani P., Priego Quesada J.I. Human body flotation and organic responses to water immersion. *Journal of Physical Education and Sport* 2013;354–61. doi: 10.22334/jbhost.v6i2.217.s47.
 15. Ofenheimer A, Breyer-Kohansal R, Hartl S, Burghuber OC, Krach F, Schrott A et al. Reference values of body composition parameters and visceral adipose tissue (VAT) by DXA in adults aged 18-81 years-results from the LEAD cohort. *Eur J Clin Nutr* 2020; 74(8):1181–91. doi: 10.1038/s41430-020-0596-5.
 16. Barwood MJ, Bates V, Long G, Tipton MJ. “Float First:” Trapped Air Between Clothing Layers Significantly Improves Buoyancy on Water After Immersion. *IJARE* 2011; 5(2). doi: 10.25035/ijare.05.02.03.
 17. Dilen DR. The Motion of Floating and Submerged Objects in the Chattahoochee River, Atlanta, GA. *J. Forensic Sci.* 1984; 29(4):11769J. doi: 10.1520/jfs11769j.
 18. Mateus M, Pablo H de, Vaz N. An investigation on body displacement after two drowning accidents. *Forensic Sci Int* 2013; 229(1-3):e6-12. doi: 10.1016/j.forsciint.2013.03.010.
 19. Bierens JJLM, Lunetta P, Tipton M, Warner DS. *Physiology Of Drowning: A Review. Physiology (Bethesda)* 2016; 31(2):147–66. doi: 10.1152/physiol.00002.2015.
 20. Madea B, Hrsg. *Rechtsmedizin: Befunderhebung, Rekonstruktion, Begutachtung*. 3. Aufl. 2015. Berlin, Heidelberg: Springer Berlin Heidelberg; 2015. Verfügbar unter: <http://nbn-resolving.org/urn:nbn:de:bsz:31-epflicht-1556609>.
 21. MSC/Circ.1046. *Guideline for the assessment of thermal protection*; 2002.
 22. John L. Hunsucker, Scott J. Davison. Time Required For a Drowning Victim to Reach Bottom. *Journal of Swarch and Rescue* 2013; (1):19–28 [Stand: 13.04.2022]. Verfügbar unter: https://journalofsar.com/wp-content/uploads/2017/01/jsar_vol1-iss1.pdf#page=22.
 23. Mateus M, Vieira V. Study on the postmortem submersion interval and accumulated degree days for a multiple drowning accident. *Forensic Sci Int* 2014; 238:e15-9. doi: 10.1016/j.forsciint.2014.02.026.