

2010-08-18

Predictive Data-Derived Bayesian Statistic-Transport Model and Simulator of Sunken Oil Mass

Maria Angelica Echavarría Gregory
University of Miami, m.echavarríagregory@umiami.edu

Follow this and additional works at: https://scholarlyrepository.miami.edu/oa_dissertations

Recommended Citation

Echavarría Gregory, Maria Angelica, "Predictive Data-Derived Bayesian Statistic-Transport Model and Simulator of Sunken Oil Mass" (2010). *Open Access Dissertations*. 471.
https://scholarlyrepository.miami.edu/oa_dissertations/471

This Open access is brought to you for free and open access by the Electronic Theses and Dissertations at Scholarly Repository. It has been accepted for inclusion in Open Access Dissertations by an authorized administrator of Scholarly Repository. For more information, please contact repository.library@miami.edu.

UNIVERSITY OF MIAMI

PREDICTIVE DATA-DERIVED BAYESIAN STATISTIC-TRANSPORT MODEL
AND SIMULATOR OF SUNKEN OIL MASS

By

Maria Angelica Echavarria Gregory

A DISSERTATION

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Coral Gables, Florida

August 2010

©2010
Maria Angelica Echavarría Gregory
All Rights Reserved

UNIVERSITY OF MIAMI

A dissertation submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

PREDICTIVE DATA-DERIVED BAYESIAN STATISTIC-TRANSPORT MODEL
AND SIMULATOR OF SUNKEN OIL MASS

Maria Angelica Echavarria Gregory

Approved:

James D. Englehardt, Ph.D., P.E.
Professor of Civil, Architectural,
and Environmental Engineering

Terri A. Scandura, Ph.D.
Dean of the Graduate School

David A. Chin, Ph.D., P.E.
Professor of Civil, Architectural,
and Environmental Engineering

Daniel Suman, Ph.D.
Professor of Marine Affairs
and Policy

Frederick Bloetscher, Ph.D., P.E.
Assistant Professor of Civil, Environmental,
and Geomatics Engineering
Florida Atlantic University

ECHAVARRIA GREGORY, MARIA ANGELICA
Predictive Data-Derived Bayesian Statistic-Transport
Model and Simulator of Sunken Oil Mass

(Ph.D., Engineering)
(August 2010)

Abstract of a dissertation at the University of Miami.

Dissertation supervised by Professor James D. Englehardt.
No. of pages in text. (198)

Sunken oil is difficult to locate because remote sensing techniques cannot as yet provide views of sunken oil over large areas. Moreover, the oil may re-suspend and sink with changes in salinity, sediment load, and temperature, making deterministic fate models difficult to deploy and calibrate when even the presence of sunken oil is difficult to assess. For these reasons, together with the expense of field data collection, there is a need for a statistical technique integrating limited data collection with stochastic transport modeling. Predictive Bayesian modeling techniques have been developed and demonstrated for exploiting limited information for decision support in many other applications. These techniques brought to a multi-modal Lagrangian modeling framework, representing a near-real time approach to locating and tracking sunken oil driven by intrinsic physical properties of field data collected following a spill after oil has begun collecting on a relatively flat bay bottom.

Methods include (1) development of the conceptual predictive Bayesian model and multi-modal Gaussian computational approach based on theory and literature review; (2) development of an object-oriented programming and combinatorial structure capable of

managing data, integration and computation over an uncertain and highly dimensional parameter space; (3) creating a new bi-dimensional approach of the method of images to account for curved shoreline boundaries; (4) confirmation of model capability for locating sunken oil patches using available (partial) real field data and capability for temporal projections near curved boundaries using simulated field data; and (5) development of a stand-alone open-source computer application with graphical user interface capable of calibrating instantaneous oil spill scenarios, obtaining sets maps of relative probability profiles at different prediction times and user-selected geographic areas and resolution, and capable of performing post-processing tasks proper of a basic GIS-like software.

The result is a predictive Bayesian multi-modal Gaussian model, SOSim (Sunken Oil Simulator) Version 1.0rc1, operational for use with limited, randomly-sampled, available subjective and numeric data on sunken oil concentrations and locations in relatively flat-bottomed bays. The SOSim model represents a new approach, coupling a Lagrangian modeling technique with predictive Bayesian capability for computing unconditional probabilities of mass as a function of space and time. The approach addresses the current need to rapidly deploy modeling capability without readily accessible information on ocean bottom currents.

Contributions include (1) the development of the apparently first pollutant transport model for computing unconditional relative probabilities of pollutant location as a function of time based on limited available field data alone; (2) development of a

numerical method of computing concentration profiles subject to curved, continuous or discontinuous boundary conditions; (3) development combinatorial algorithms to compute unconditional multimodal Gaussian probabilities not amenable to analytical or Markov-Chain Monte Carlo integration due to high dimensionality; and (4) the development of software modules, including a core module containing the developed Bayesian functions, a wrapping graphical user interface, a processing and operating interface, and the necessary programming components that lead to an open-source, stand-alone, executable computer application (SOSim – Sunken Oil Simulator).

Extensions and refinements are recommended, including the addition of capability for accepting available information on bathymetry and maybe bottom currents as Bayesian *prior* information, the creation of capability of modeling continuous oil releases, and the extension to tracking of suspended oil (3-D).

Keywords: sunken oil, Bayesian, Gaussian, model, stochastic, emergency response, recovery, statistical model, multimodal.

TABLE OF CONTENTS

LIST OF FIGURES.....	v
LIST OF TABLES.....	viii
Chapter	
1 INTRODUCTION.....	1
1.1 Motivation.....	1
1.2 Literature Review and Related Work.....	5
1.3 Objectives of this Dissertation.....	13
1.4 Scope of this Dissertation.....	16
1.5 Organization of this Dissertation.....	18
2 METHODS.....	20
2.1 Predictive Bayesian Multimodal Gaussian Model Development.....	23
2.2 Development of Methods for Accounting for Approximately Curved Boundary Conditions.....	30
2.3 Development of Algorithms for Integration over Highly Dimensional Uncertain Parameter Space.....	31
2.4 Development of the Computer Application for the Model.....	32
3 PREDICTIVE BAYESIAN DATA-DERIVED GAUSSIAN MODEL OF SUNKEN OIL MASS.....	34
3.1 Introductory Remarks.....	34
3.2 Predictive Bayesian Gaussian Approach.....	35
3.3 Assumed or Specified Parameter Domains.....	44
3.4 Projecting the Oil Mass in Time.....	48
3.5 Integration to Obtain the Predictive Relative Concentration Profile...	56
3.6 Verification of the Functionalities of the Model.....	58
3.7 Assessment of the Physical Parameter Domain.....	74
3.8 Discussion and Importance of Results.....	79
4 COMPUTING LAGRANGIAN RELATIVE CONCENTRATION PROFILES SUBJECT TO CURVED, CONTINUOUS OR DISCONTINUOUS REFLECTING BOUNDARY CONDITIONS.....	82
4.1 Overview.....	82
4.2 Methods and Background.....	83
4.3 Discussion of the Method of Images to Provide Predictive Gaussian	

	Distributed Sources with Variable, Continuous or Discontinuous, Irregular, Approximately Curved Boundary Conditions.....	88
4.4	Conclusions.....	115
5	SOSIM SYSTEM DEVELOPMENT, INCLUDING A COMBINATORIAL ALGORITHM FOR THE COMPUTATION OF MULTI-MODAL, HIGHLY DIMENSIONAL STOCHASTIC FUNCTIONS.....	116
5.1	Python: The Programming Language.....	116
5.2	Methodology for Software Development.....	118
5.3	Algorithm Development in SOSim Modules.....	120
5.4	Software Operation.....	141
5.5	Discussion and Importance of Results.....	143
	CONCLUSIONS.....	144
	WORKS CITED.....	148
	APPENDICES.....	152

LIST OF FIGURES

Figure

1.1	Principal causes of oil spills in the world, Source: ITOPF, 2009.....	6
1.2	Case-based fractions of oil spilt that undergo processes in the seawater and the atmosphere. Source: Cedre, 2009.....	7
2.1	Conceptual map for the development of the sunken oil mass simulator SOSim.....	20
2.2	Aerial plot presenting the results of a sampling campaign following the 2005 DBL 152 spill near Port Arthur, TX.....	28
3.1	Distribution of physical variability assuming perfect information, or no uncertainty, and predictive Bayesian distributions based on limited data, and limited data with professional judgment, showing the decrease in information entropy with increased information availability.....	36
3.2	Graphical depiction of the physical idealization of the boundary conditions to apply the Method of Images in the SOSim model.....	53
3.3	Aerial plot presenting the results of a sampling campaign following the 2005 DBL 152 spill near Port Arthur, TX. The shaded area contains the portion of the data used to confirm the ability of the model to locate sunken oil.....	61
3.4	Zoom of the shaded area of Figure 3.3, showing the recorded relative concentrations input to SOSim and their geographical location used to confirm the ability of the model to locate sunken oil.....	62
3.5	General location of the DBL 152 spill at 29.205° N, 093.4683° W.....	63
3.6	Relative probability of finding sunken oil 12 hours after the spill.....	64
3.7	Relative probability of finding sunken oil 17.5 days after the spill.....	65
3.8	Relative probability of finding sunken oil 19.5 days after the spill.....	66
3.9	Synthetic data on relative sunken oil concentrations in percent, for samples assumed collected on two different days, (1) 6 days after the spill, and (2) 10 days after the spill.....	69
3.10	General location of the simulated spill scenario of Scenario 2.....	70
3.11	Relative probability of finding sunken oil 8 days after the spill (1 day after the first sampling campaign).....	71
3.12	Relative probability of finding sunken oil 11 days after the spill (5 days after the first sampling campaign and 1 day after the second sampling campaign), updated based on the second data set.....	72

3.13	Relative probability of finding sunken oil 13 days after the spill (3 days after the second sampling campaign).....	73
3.14	Assessment of the likelihood of the observed limited data of the DBL-152 case as function of valid parameter combinations (θ_k) within the parameter space (Θ).....	75
3.15	Graphical representation of the superimposed Gaussian distributions that maximize the likelihood function obtained by SOSim for the DBL 152 limited data campaign.....	78
4.1	Sectional views of a discharging well near an impermeable boundary and the equivalent system using the method of images (Todd and Mays, 2005).....	84
4.2	Typical 2-D layout of an image well system used in hydraulics of groundwater, where a is the perpendicular distance from the source to the vertical boundary and b is the perpendicular distance to the horizontal boundary.....	85
4.3	Plan of an image-well system for a rectangular aquifer (Adapted from Todd and Mays, 2005).....	86
4.4	Image well system for a discharging well in an aquifer bounded by two impermeable barriers intersecting at 45° (Todd and Mays, 2005).....	87
4.5	Graphical depiction of the Method of Images in one dimension applied to sunken oil distributions.....	89
4.6	Reflection of a Gaussian patch across a sloped single line segment boundary.....	92
4.7	Geometry of the reflection of the Gaussian mean with respect to a single line boundary.....	93
4.8	Algorithm to find the Gaussian mean of the imaginary Gaussian puff, (μ_{x_r}, μ_{y_r}) , from the mean of the source and two points of a line traced on the single boundary.....	94
4.9	Approximation to the relationship between source and imaginary standard deviations.....	95
4.10	Set of bi-dimensional Gaussian profiles showing correlation coefficients between x and y directions. Note that the correlation coefficient reveals the direction and noisiness of the linear relationship (figure adapted from Wikipedia.org).....	96
4.11	Relative directions of Gaussian source and single line segment boundary, used to find the parameters of the imaginary Gaussian puff.....	97
4.12	Algorithm to find the correlation coefficient of the imaginary Gaussian puff, ρ_r , from the slope of the single line segment boundary, m , and the correlation coefficient of the Gaussian source, ρ_s	98

4.13	(a) A geographic land mass to the northwest of the Peninsula of Florida, on the Gulf of Mexico, is approximated by a set of continuous non-horizontal and non-vertical lines, joined by circular vertices. (b) Zoomed in bay that can be represented by a polyline of only 10 vertices.....	99
4.14	Geographic scenarios that can be approximated by polylines. Vertices are shown as circles. (a) Bay in Louisiana State, (b) the Caribbean waters between the southernmost point of Florida and the northern part of Cuba...	100
4.15	Hypothetical approximation to the reflection of a bivariate Gaussian distribution with respect to multiple, continuous line boundaries.....	102
4.16	Geometry of the first-order reflection of the Gaussian mean with respect to multiple line boundaries.....	107
4.17	Example geometry of one of the second-order reflection cycles to obtain imaginary Gaussian means with respect to multiple line boundaries.....	108
4.18	Iterative algorithm to control the production of the Gaussian means of the successive line-based and ordered reflections.....	109
4.19	Iterative algorithm to control the production of the correlation coefficients of the successive line-based and ordered reflections.....	111
4.20	Relative mass distribution among ocean and the imaginary land counterpart for a Gaussian source that hits the coastline shown in (a), approximated by polylines when (b) no boundary conditions are in effect, (c) after accounting for zero and first-order reflection processes, and (d) after accounting for zero, first and second-order reflection processes...	114
5.1	SOSim composition and interaction between modules.....	121
5.2	Algorithm for conversion of coordinate systems.....	126
5.3	Representation of the superposition concept in three dimensions.....	134

LIST OF TABLES

Table

3.1. Default parameter ranges of the model.....	46
3.2. Input and output for Verification Scenario 1: DBL 152.....	59
3.3. Input and output for Scenario 2.....	67
3.4. Parameter set that produces the maximum likelihood of being observed given the parameter space (Θ) for the limited data obtained of the DBL-152 oil spill.....	76

Chapter 1. Introduction

1.1 Motivation

The ultimate impact of an oil spill stems from several factors, including its size, geographical location, political and practical context of the response, as well as the occurrence of suspended or sunken oil. Sunken oil can occur following a spill of heavy oil, or of lighter oil that entrains sediment or loses its light constituents in the short-term weathering process. The term sunken oil is used in this dissertation to refer to oil on the bottom, though some of the discussion and approach may also apply to oil suspended in the water column. As noted by the U.S. NOAA's Coastal Response Research Center (CRRC, 2007), "In the past few years, spills of non-floating oil and oils that become submerged as a function of sediment entrainment have presented significant response challenges and have resulted in enormous dollar-per-barrel recovery costs. Currently, the ability to forecast submerged oil movement, estimate water column concentrations of large droplets, and efficiently recover sunken masses in an operationally expedient way is quite limited. Additionally, as this category of oil is difficult to locate, track, and retrieve, managers have difficulty maintaining public confidence with regard to response termination." Problems in locating and tracking sunken oil are further exacerbated by the expense of developing and deploying new remote sensing techniques, and because some oils may re-suspend and sink with changes in salinity, sediment load, and temperature.

Many models have been used to simulate the drift and fate of oil slicks in one, two, or three dimensions using Eulerian or Lagrangian modeling techniques (Ojo *et al*, 2007;

Spaulding *et al*, 1994; Spaulding *et al*, 1997; Beegle-Krause, 2001; Yapa, 1994; Sugioka *et al*, 1999). The spill-trajectory model developed and used most recently by the U.S. National Oceanic and Atmospheric Administration (NOAA) is the General NOAA Oil Modeling Environment (GNOME), developed by the Hazardous Materials Response Division (HAZMAT) Office of Response and Restoration. GNOME includes statistical capability in the form of Best Guess and Minimum Regret solutions (Galt, 1998). However, existing models are not developed for the detection and mapping of sunken oil. Other possible approaches to locating and tracking sunken oil include electro-acoustic detection, mechanical detection, and inspection by divers, as summarized in the Technical Guidelines on Sunken Oil Assessment and Removal Techniques (NOAA, 2006). However, integrated models for short and long-term sunken oil tracking with on-scene calibration capability during emergency response have been identified as research needs (CRRC, 2007).

The occurrence of sunken oil is difficult to predict in time and space before, during, and after cleanup, using either contaminant transport models or field data, for two reasons. First, hydrodynamic or particle tracking models may be difficult to deploy and calibrate for tracking of sunken oil, due to the site-specific and potentially transient nature of sunken oil occurrence and location under changing field and oil conditions, and limitations in the available information on advective/dispersive forces acting on sunken oil on the bottom. Second, the collection of field data on sunken oil locations is intrinsically expensive. Modeling techniques that accept near-real time field data as input

and account quantitatively for both uncertainty and variability are not well-developed, and have not been available to support response, cleanup, and damage assessment decisions.

The use of Bayesian modeling techniques to incorporate non-numerical types of information in probabilistic assessments has exploded in recent years due to the development of new computational approaches (e.g. Markov Chain Monte Carlo). The approach allows integration of available numerical data together with any less-quantitative information, with rigorous accounting for uncertainty in accordance with the laws of probability. Therefore, it represents an approach to exploiting available field data on sunken oil locations in time and their intrinsic physical characteristics and parameters together with statistical tools that make the prediction of future states possible. The Bayesian approach involves inclusion of non-numeric or numeric relative information to develop *posterior* probability distributions for uncertain quantities of interest. The intrinsic physical information contained in the collected field data along with the overall information about the spilt oil, the accident location, and a set of elapsed times are used to calibrate a spill scenario, that is, to obtain a posterior probability distribution. A posterior probability distribution can be used in predictive Bayesian theory to foretell the behavior of another possible observation without the need of performing further experimental sampling on the zone of possible observation. The result is an “extrapolated” prediction, an unconditional relative probability distribution that is wider than the posterior distribution because it now contains a degree of uncertainty. The

physical mechanisms driving the new (predictive) distribution are the same that had been able to develop the physical state and distribution of the sampled data set from the time of the spill until the sampling campaign time. The approach can be applied to continuously update calibrations so that predicted states reflect the most current physical changes and intrinsic uncertainties and variability induced by the environment on the measured data.

Predictive Bayesian methods involve the development of unconditional probability distributions for the quantity of interest, by integrating over all physically-possible values of the uncertain parameters. The approach has been used in oil spill prevention and preparedness planning (Obie and Englehardt 1996; Douligieris *et al.*, 1998) and for other applications including hurricane, environmental, health, and safety risk analysis (Aitchison and Dunsmore, 1975; Englehardt, 2004, 1995; Bloetscher *et al.*, 2005; Englehardt and Swartout, 2004; Englehardt *et al.*, 2003; Anex and Englehardt, 2001; Englehardt and Peng, 1996). For example, a predictive Bayesian compound Poisson model was developed to forecast changes in oil spill volumes onshore in the Gulf of Mexico in response to proposed changes in oil transportation and response equipment and policies, given geographically-defined historical oil spill data, shipping routes and volumes, and hydrodynamic modeling results (Obie and Englehardt, 1996; Douligieris *et al.*, 1998). However, the approach has not been brought to bear on the problem of locating sunken oil.

1.2 Literature Review and Related Work

The expression “oil spill” refers to a violent spillage of hydrocarbons concentrated in a specific area, surpassing the natural assimilation capacities of the surrounding environment. Sinking is the physical mechanism by which oil masses that are denser than the receiving water are transported to the bottom. The oil itself may be denser than seawater, or it may sink by chemical or physical means: chemically, the soluble fractions separate from denser fractions, leaving the latter with the capability of sinking; physically, oil may have incorporated enough sand to become denser than the water and therefore sink.

1.2.1 Occurrence of Sunken Oil

Vast quantities of crude oil and refined products are transported over long distances, incurring constant and substantial risks of accidents. In particular, roughly one half of the oil consumed worldwide is transported by sea, in ~9,130 oil tankers counted worldwide (ITOPF, 2009). High vessel densities in maritime routes and loading/offloading ports, longer journeys, and intricate aspects of geographical location all increase the risk of oil spills due to collision, negligence, grounding, or defects in a vessel’s structure. Insurance statistics indicate that most oil tanker accidents resulting in marine oil spills result from human error, including “... badly handled maneuvers, neglect maintenance, insufficient checking of systems, lack of communication between crew members, fatigue, or an inadequate response to a minor incident...” (Cedre, 2007). From a more practical point of

view, examination of the circumstances surrounding accidents (ITOPF, 2004) indicates a high percentage of spills due to groundings and collisions. Figure 1.1 shows statistics on the reasons of oil spill occurrences.

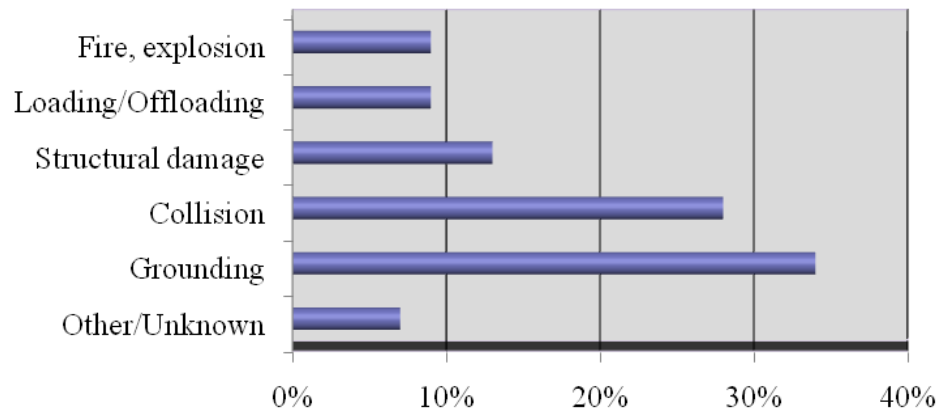


Figure 1.1. Principal causes of oil spills in the world, Source: ITOPF, 2009.

The nature of oil transported by sea varies from the lightest oil (highly volatile hydrocarbon or gasoline) that floats on the sea water and evaporates to the atmosphere, to heavy fuel oils, of which perhaps only 10% will evaporate (ITOPF, 2009) with much of the remainder sinking to the bottom. Processes that affect sinkable oil mass following a spill include evaporation, water and oil mixing and sedimentation. Figure 1.2 shows case-based examples of the approximated fractions of spilled oil mass that undergo evaporation and all other processes in seawater.

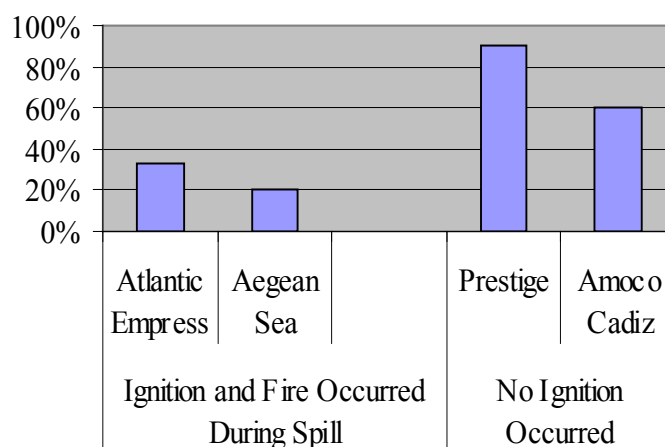


Figure 1.2. Case-based fractions of oil spilt that undergo processes in the seawater and the atmosphere. Source: Cedre, 2009.

In general, crude oils and certain heavy refined products and sludge deposits have the potential to sink, with sinking of oil generally becoming important from 1-8 days following a spill (Coastal Response Research Center, 2007). Sunken oil then accumulates on the bottom in reported thicknesses of about 2.5 inches, regardless of the size of the oil patches (Beegle-Krause *et al.*, 2006). Sedimentation of thick and heavy oil occurs rapidly, and involves the majority of the oil minus a small percentage (< 10%) that evaporates and a portion that remains in bubbles in the water column for a short time. Evaporation and mixing with water diminish the sinkable fraction, while sediment entrainment increases sedimentation. Incorporation of water into the oil mass due to mixing may prevent sinking for periods of hours to days following the spill. For spills of light oil, sedimentation generally occurs over a long period of time and involves < 5% of the oil mass (Cedre, 2007).

Sinking of heavy oils (API gravities less than ~ 7.0) due to gravity is more likely in quiescent seawater where currents are under 0.1 knot (Research Planning Inc., 2001) because higher currents typically keep oil droplets suspended in the water column longer. According to NOAA event comparison charts, oils denser than local water have not been observed to impact the shoreline unless the source of the oil was within the surf zone or the oil moved into relatively denser water and thus became buoyant (Beegle-Krause *et al.*, 2006). From Protocols for NRDA Surveys (Research Planning Inc., 2001), "... sunken oil can be buried by silt in harbors or sand in offshore areas within days to weeks. Once buried, it can remain for years, only to be exposed by storms or dredging operations." If near shore, sunken oil and tar can wash up on beaches following storms for years following a spill. Gravity will induce flow of the oil in the offshore direction, towards deeper water (Beegle-Krause *et al.*, 2006).

As reported in Protocols for NRDA Surveys (Research Planning Inc., 2001), heavy oil can temporarily accumulate in low-flow zones. In rivers, accumulation may occur in backwaters, sloughs, inactive scour pits, and in the lee of point bars, wing dams, and other man-made obstructions. In estuaries, potential accumulation areas include man-made depressions (e.g., dredged channels, marinas and boat slips, prop scour pits, turning basins), natural scour pits active during periods of high flow, and abandoned channel meanders. Along the outer coast, accumulation may occur in troughs between offshore

bars, lagoons or pools protected by offshore rocks or coral, reef flats protected by reef crests, and in the lee of any obstruction of currents along the coast (e.g., rocks, jetties, and breakwaters).

Oils lighter than the receiving water may sink by (a) adhering to sand-sized particles during mixing in the surf zone; (b) stranding on shore, picking up sand, then being eroded from the beach by waves and deposited in the near-shore zone; and (c) adhering to the substrate during low water, then not re-floating when water levels rise. The latter mechanism is more likely in rivers and streams where water levels may fluctuate. In general, sunken oil that is intrinsically lighter than surrounding water (i.e., that sank by mixing with sand) can re-float if the oil separates from the sand or bottom substrate. Such separation may occur upon warming, due to the reduction in viscosity.

Sedimentation is of significant concern even if the oil has a light density, if a spill occurs in a nearshore environment where the oil can mix with sand or sediment causing it to sink, as happened in the Braer, Erika and Prestige incidents (ITOPF, 2009). In such cases the sunken oil is extremely difficult to detect and recover. In general, once on the bottom, most hydrocarbons easily enter gaps and flow by gravity so deep that it may be impossible to find by inspection.

Sunken oil mats tend to remain stationary in the absence of storms; local bottom currents typically do not have enough energy to move the mat. Release and resuspension of parts of submerged oil mats by long period gravity wave energy occurs most often

along the near-shore shelf where water is shallower and wave energy extends closer to the bottom depth (Dean and Dalrymple, 1991). As a result of this increasing tendency to remain stationary following deposition in deeper water, the mats tend migrate in the offshore direction over time (Beegle-Krause *et al.*, 2006). If the mat is broken up into particles during mixing, it does not tend to re-coalesce. There is a significant difference in the oil content of oil masses that sink by gravity and those that sink by entrainment and sedimentation. The former may contain only a few percent sediment, whereas oil-contaminated sediments accumulated on the seafloor generally contains less than 1% oil (Research Planning Inc., 2001).

Hydrocarbons that are not removed from certain ocean bottoms can seriously damage populations living within the sediment substrate. Sunken oil weathers slowly; therefore toxic components can persist and be a source of exposure during re-floatation or benthic transport. A spill of heavy fuel oil is likely to cause much more damage than a crude oil spill of a corresponding size. The duration of spillage also plays an important role. A sudden violent release will concentrate the effects on a smaller area as compared with a long, slow leak.

Several properties affect the classification of oil as heavy or light, and consequently can modify the propensity of the oil to sink as explained above. These combined properties, including density, viscosity, pour point, solubility, chemical composition and potential for emulsification, along with associated short-term behavior in the environment and impacts to natural resources (Research Planning Inc., 1994), allowed for the

classification of oil into six broad categories. In general, Type 1 oils are very light, including gasoline and very volatile hydrocarbons. Type 2 are moderately volatile and soluble, including jet fuels, diesel fuel, number 2 fuel oil, and light crude oils. Type 3 oils include most crude oils, known by their persistence and diminished propensity to evaporate (about one third of the total mass evaporates within 24 hours). Type 4 oils may have little propensity to evaporate or dissolve, and high likelihood of sinking. Type 5 oils have essentially no evaporation potential, weather very slowly, and sink immediately, including heavy industrial fuel oils. Type 6 oils include heavy animal or plant oils. This classification of oils is tightly related to API gravity, a measurement of the relative density of petroleum liquids developed by the American Petroleum Institute and adopted by the oil industry worldwide. Type 1 oils may have API gravities around 31 °API, whereas a Type 4 oil can have an API gravity of less than about 10 °API, in which case the oil will typically sink in water.

1.2.2 Transport and Accumulation

The following points concerning characteristics of the initial fate and transport of sunken oil have been excerpted, adapted, interpreted, and/or condensed from the references cited:

- After events of high bottom energy, sunken oil can be resuspended and sometimes mixed until it is broken up into small globules. These smaller globules of heavy oil are not expected to coalesce into a larger slick at some later time (Beegle-

Krause *et al.*, 2006), but rather to weather and degrade over long periods. At depth, the so-called “convergence zones” found in the surface are not found, and a mechanism for bringing the globules together no longer exists in the bottom (Beegle-Krause *et al.*, 2006);

- Long-term transport of heavy oil is seldom compared to long-term sediment transport on continental shelves. Events with sufficient energy are more likely to be caused by long-period waves than by the local bottom currents (Beegle-Krause *et al.*, 2006); and
- Sediment is typically transported greater distances along the shelf than across the shelf (Beegle-Krause *et al.*, 2006). This observation could represent prior information for the development of Bayesian prior probability distributions for coefficients of advection and dispersion in directions perpendicular and parallel to shore.

1.2.3 Mechanisms of Resuspension of Sunken Oil

When a high-density oil spill occurs, a large portion of the oil will sink to the bottom to form large discrete mats in many areas and smaller globules in others (Beegle-Krause *et al.*, 2006). Here, the term “globule” is equivalent to “tarball”, also used in the literature. Observational data in NOAA data bases suggest that oil remains in areas of heavier concentration until high-energy storms redistribute the oil. The following are excerpted or adapted from Beegle-Krause *et al.* (2006), except as noted:

- Literature suggests that average current conditions will not be sufficient to move the oil in a continuous manner (Beegle-Krause *et al.*, 2006; Boehm *et al.*, 1981). Rather, “The oil will remain stationary on the bottom until an event occurs with enough energy to stir it up into the water column;”
- “Tarmats occur when floating oil moves into the surf zone, collects sediment, and sinks;”
- “With enough energy, tarmats in the bottom generally break up into smaller pieces of oil that spread out into a large area. Otherwise, the tarmats remain stationary and intact;”
- “Outside the inner shelf, where coastal current jets form, long-period waves will be the only source of turbulent energy at the bottom other than large storms strong enough to mix the entire water column;”
- “The oil will behave similarly to local sediments in terms of episodes of burial and re-exposure and mobilization into the water column;”
- The energy required to move the oil varies with depth and site and is unknown. A break-up energy level has been estimated as $6 \text{ m}^2/\text{Hz}$ (Beegle-Krause, *et al.*, 2006).

1.3 Objectives of this Dissertation

Given the site-specific nature of the occurrence of sunken oil and the need to project its location in time, a statistical data-limited technique representing a cross between a

statistical static sampling plan and a contaminant transport model was proposed for development in this research. General objectives of this dissertation include:

1. To develop the first pollutant transport-statistical model to our knowledge capable of computing unconditional probabilities of relative pollutant mass as a function of time and space;
2. Develop a method of computing Lagrangian relative concentration profiles subject to variable, continuous or discontinuous, irregular, approximately curved boundaries in two dimensions applied to Gaussian-distributed sources;
3. Develop a combinatorial algorithm for the computation of multi-modal, highly dimensional functions such as likelihood functions, posteriors, conditionals, and ultimately predictive Bayesian unconditional functions when the dimensionality of the problem prevents both the use of Monte Carlo Methods and matrix algebra in computer algorithms
4. Develop an open-source computer application (SOSim – Sunken Oil Simulator) with graphical user interface, capable of calibrating instantaneous oil spill scenarios using limited data, obtaining maps of relative probability profiles as processed by the transport-statistical model, at different prediction times and user-selected geographic areas and resolution, and capable of performing post-processing tasks proper of a basic GIS-like software.

To meet current needs in terms of locating sunken oil during emergency response operations, it was determined that information on bottom “currents” and their potential

forcing of the movement of sunken oil was too limited to be a primary source of input information and driving mechanism for the model (Beegle-Krause *et al*, 2006). Rather, it was desirable to use the limited field data that is gathered after an oil spill. Among specific objectives of the stand-alone, open-source, executable model with graphical user interface (GUI), the following capabilities are included:

- Assessing sunken oil locations based on irregularly-sampled, certain limited available physical data collected shortly after a spill event,
- Projecting oil location in time based on subsequent limited field data that intrinsically carried updated physical information on parameters and variability, Providing updated predictions based on additional, relative field data, from possibly different and irregular geographical areas, as they become available.
- Presenting unconditional probabilities of sunken oil in output maps that belong each to a user-requested prediction time,
- Accounting for the time lapse before depositing on the bottom that an oil may experiment due to its potential for sinking and short-term weathering,
- Multimodality, which is the capability of the predictive model to infer from the data whether or not the sunken oil is distributed in single or multiple patches, and to track and predict this multimodal behavior in time.

Specifically, a predictive Bayesian multimodal Gaussian model of sunken oil locations across a relatively flat-bottomed bay was proposed, to accept possibly irregularly-sampled field data at times shortly after each spill event when sunken oil has appeared on the bottom.

The calibrated model also was to have capability for accepting input in the form of ranges of possible parameter values (not prior distributions on the parameters or expected values) based either on hydrodynamic data or professional judgment, to over-ride default ranges assigned in the research. In that way, physical information contained in the limited field data can be integrated with certain other assumed or measured ranges of parameters to assess and project the location of dispersing sunken oil masses both before and after cleanup begins, for spill response and decision-making, cleanup, and damage assessment. Finally, the model should be capable of being enhanced in the future to accept bathymetric information and other means of prior information as input.

1.4 Scope of this Dissertation

The scope of the research was limited to:

- Sunken oil;
- Relatively flat bay bottoms, dredged bays, reef flats and lagoons or pools protected by offshore rocks; bays with steeply sloped bottoms would require

capability for the use of bathymetric data as prior information, a possible future enhancement;

- Resolution down to the scale of the tidal excursion (oil locations effectively averaged across this excursion);
- Discrete accidental oil releases (as opposed to natural, progressive oil seepage);
- Relatively uncomplicated continuous concave and convex shoreline geometries (although the method was also developed for discontinuous boundaries);
- Out of the scope of the research is modeling in straits, inland water bodies, harbors, islet areas, and like geographies that are not addressed due to computational complications and the sometimes transient nature of small-scale features.

This dissertation describes the development of the Sunken Oil Simulation (SOSim) model, to be used for identifying sunken oil hotspots, tracking sunken oil following a spill, targeting cleanup activities, and supporting cleanup termination decisions. The model and the software represent a new approach to pollutant tracking by inference from limited field data alone. The dissertation also explains the design of a method developed and used to compute Lagrangian relative concentration profiles subject to curved, continuous or discontinuous boundary conditions; and explains the combinatorial

algorithm established for the calculation of likelihood functions, conditional functions, and predictive unconditional functions; subject to high parameter dimensionality.

The final product of the research is a computer executable stand-alone software package created and tested under Microsoft Windows 32 bit operative system, programmed in the Python language (“Python is often compared to Tcl, Perl, Ruby, Scheme or Java”(Python Software Foundation, 2010)) and embedded in its own GUI. Included in this dissertation is a user’s manual for the open-source product (Appendix A). Because the model was developed using only open source software and GPL derivatives, its use is governed under the GPL (General Public License) license terms (license terms included in the source code). This tool will allow the response coordinator to choose and customize response options, including actions at the source, at sea, approximated solutions near the shore, plan operations at predetermined times and locations based on projected sunken oil locations, and plan the overall cleanup and recovery phase to mitigate impacts.

1.5 Organization of this Dissertation

First, the objectives and scope of this dissertation are unveiled. Following in this first chapter, current information reported in the literature and relevant to the transport of sunken oil is reviewed. Chapter 2 outlines the methodologies employed in the research for the development of each objective. Based on the information of Chapter 1; a conceptual model for the predictive Bayesian model, and its analytical expressions, are

presented in Chapter 3. Chapter 3 also describes the functionalities of the model, the statistical approach to the development of SOSim, and the confirmation of the functionalities developed using limited real and synthetic data on sunken oil location, with conclusions regarding further testing and verification through the application of the model as a predictive, decision-making tool. Chapter 4 contains a complete description, discussion and testing of the method developed to address approximately curved boundary conditions under the light of the method of images. Chapter 5 presents the computational aspects of the Sunken Oil Simulator SOSim divided in 3 parts that correspond to each of the Python modules created for the software package. The new numerical methodology developed to solve the stochastic integral equation of the analytical Bayesian model subject to its high dimensionality is explained in detail in this chapter as it belongs to the core module of the SOSim package. The user manual of the computer application is included as an appendix in this dissertation.

Chapter 2. Methods

The conceptual map that summarizes the procedure and organization for developing the predictive sunken oil mass simulator SOSim is shown in Figure 2.1.

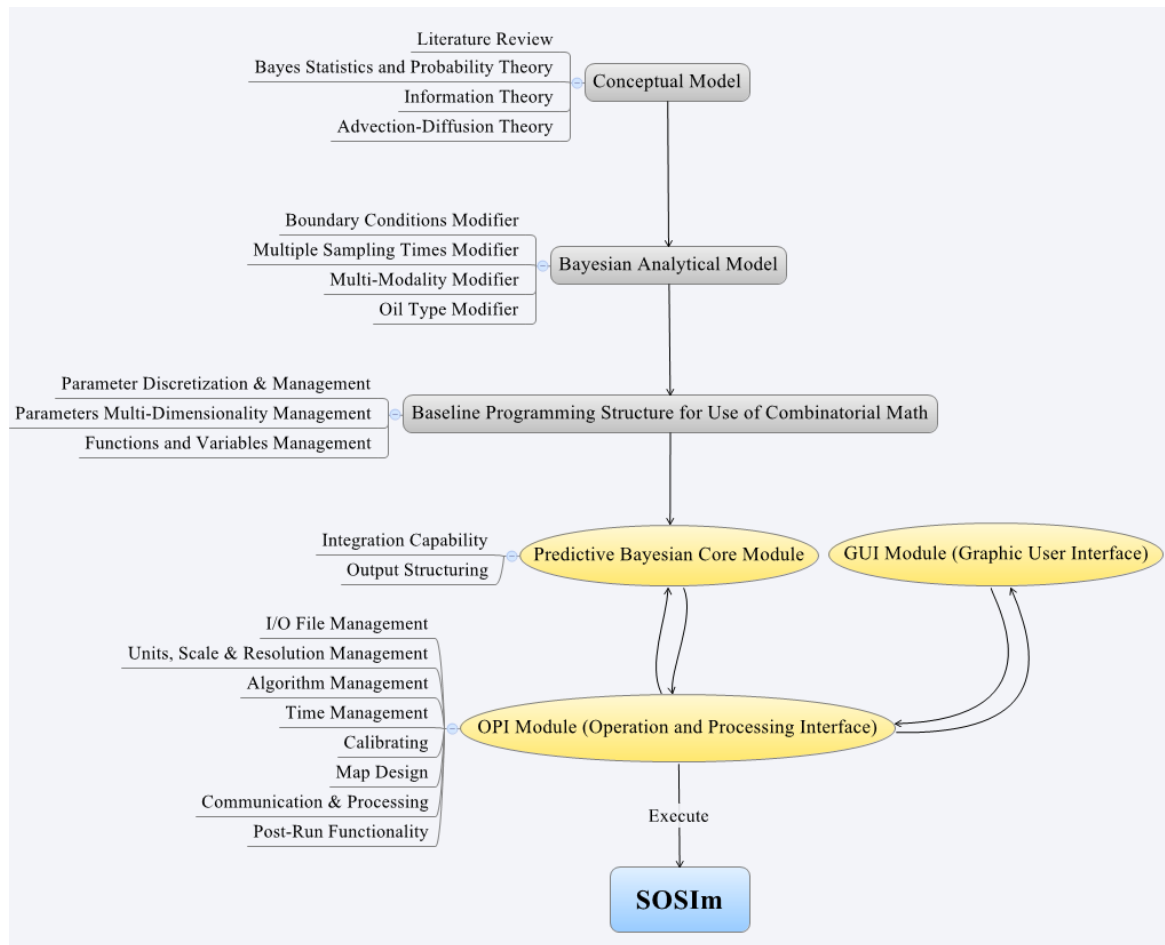


Figure 2.1. Conceptual map for the development of the sunken oil mass simulator SOSim.

The literature review developed in Chapter 1 was used together with the methods described in this chapter to develop the conceptual model targeting end user needs as named in the introduction, objectives, and scope of the research. The initial concept was a

predictive Bayesian superimposed 2-D Gaussian model incorporating computer algorithms to allow estimation of the highly parameterized model given limited, relative, subjective numerical data and possibly additional information on the surroundings after the oil has begun to sink. The model was to be capable of accepting limited available field data on oil spill concentrations, whether quantitative or qualitative in nature, sampled randomly in space, from multiple sampling campaigns, and to include default integration domains for advective velocities and coefficients of diffusion in two horizontal directions. Model input would include, other than general integration domains, the time, location and type of oil of the spill, location of all oil concentration data, and average time of sampling campaigns.

The methods used are described in the chapter according to four identified areas of need, as follows:

1. Predictive Bayesian Multimodal Gaussian Model Development

Development of the multi-modal predictive Bayesian Gaussian transport model included:

- a. Development of an understanding of the factors relevant to the modeling of sunken oil fate and transport, particularly as related to the bathymetric characteristics of bays for which a Gaussian modeling approach would be applicable;

- b. Development of analytical expressions that simulated the proposed modeling environment, comprising proposed capabilities such as multimodality and superposition, transitivity, and time and location multiplicity of sampling campaigns;
 - c. Use of the method developed for computing Lagrangian concentration profiles subject to curved continuous boundary conditions;
 - d. Use of the computational approach to be developed as an additional objective, that would allow a highly parameterized predictive Bayesian multimodal Gaussian model of sunken oil location, within the correct physical parameter domain, to be executed on a computer;
 - e. Verification of the functionalities developed for the model using partial real and synthetic sunken oil location filed data sets.
2. Development of Methods for Accounting for Approximately Curved Boundary Conditions

Development of a method of computing Lagrangian relative concentration profiles subject to curved, continuous or discontinuous reflecting boundary conditions; included:

- a. Consideration of the approaches to modeling boundary conditions in two dimensions, subject to the variable geometry of different modeling scenarios, and

- b. Understanding and applying the mechanisms to guarantee oil mass conservation within the bay at all times and modeling scenarios.
3. Development of Algorithms for Integration over Highly Dimensional Uncertain Parameter Space

Development of the combinatorial algorithm for the calculation of likelihood functions, conditional functions, and predictive unconditional functions when the high dimensionality of the problem precludes both the use of Monte Carlo methods or matrix algebra in computer algorithms.

4. Development of the Computer Application for the Model

Software development, including the creation of modules to gather user information, to graphically interface with the user, to process input, to operate and control algorithms, was performed. The software is an open-source, stand-alone (when not executed from source), executable computer application (SOSim – Sunken Oil Simulator) that includes a GIS-like graphical user interface developed from scratch.

2.1 Predictive Bayesian Multimodal Gaussian Model Development

This section describes the different methods used to trace the path of development of the multi-modal, data-derived predictive Bayesian maximum likelihood model of sunken oil mass, without detailing the methodology for boundary condition modeling and

computer algorithms methodology, which pertain to the methods of objectives 2 and 3 (explained in detail in Chapters 4 and 5).

2.1.1 Understanding the Factors Relevant to the Modeling of Sunken Oil Fate and Transport

The methodology to pursue this objective was the gathering, compilation and synthesizing of data on the occurrence and transport of sunken oil as reported for previous spills and as described in the specialized literature; and the later selection and maturing of the concepts that would be useful within a Bayesian approach. Steps towards the objective allowed for: (a) the evaluation of possible methodologies and approaches, (b) understanding the processes of hydrodynamic fate and the transport governing the behavior of sunken oil mass as a basis for development of the conceptual model and specification of default possible values and/or ranges of possible values of model parameters, and (c) the determination of the appropriate geographical scale and resolution for the model.

2.1.2 Methods for Developing the Analytical Expressions that Simulated the Proposed Modeling Environment

Following the conceptual model, the arrangement of the mathematical model was possible by means of understanding the mechanics of the Bayes probability theory and knowing the implications of crossing multimodal Gaussian Bayesian statistical parameters with hydrodynamic concepts and, in detail, with the advection-diffusion

theory and its transient parameters. Methods of the information theory were also used for the mathematical model, as explained later in this dissertation, in particular for selecting the form of the statistical error function that should be part of the joint probability of oil concentrations given the unknown parameter space (the likelihood function).

For the optimum multi-modal predictive Bayesian Gaussian statistical model of sunken oil locations across a bay that would produce assessments of sunken oil locations in time; input data for the mathematical model and computer algorithms were to include dates of spill occurrence and sampling campaign(s), field data collected after oil has begun to sink at and around a spill site (which would intrinsically include, related to the spill location and time, information on the physical driving forces, modeled using Fickian transport assumption), and approximated boundary conditions if applicable for the desired modeling zone. Algorithms were to infer Bayesian posterior probability distributions for uncertain model parameters describing the dispersion and movement of sunken oil patches in time.

The multimodal aspect of the Gaussian model was needed to accommodate oil partitioning in globules and mats as described in the specialized literature; also, to account for oil accumulating in multiple areas of the bay as a result of localized sediment entrainment, localized bathymetric catchment areas (this last works only as function of the location of the sampled data, not as predicted bathymetric accumulation given that bathymetry as prior information has not been incorporated), and other effects. The multimodal capability was provided by superimposing multiple 2-D Gaussian *patches* in

the model. Each patch was assigned a weighting parameter, representing the fraction of the total sunken oil contained in that patch, with all fractions summing to an arbitrary constant value of unity representing the unknown total sunken oil mass. Because the total mass of sunken oil was not expected to be known, the output of the model is referred to as *relative* oil mass. Thus, the traditional requirement for normalizing the area under the Gaussian distributions to a value of unity was not necessary and was relaxed. Because of the lack of a need to normalize individual Gaussian distributions, there was no need to normalize the likelihood functions used to develop the Bayesian posterior distributions. Therefore, Markov chain Monte Carlo (MCMC) computation was not required. Because of the computational demands of MCMC methods, the likelihood that the approach would not be successful for such a highly-parameterized model (23 parameters), and the relatively unskewed and regular nature of the parameter distributions to be integrated, it was decided to use a straightforward Riemann sum after a combinatorial methodology to obtain the final predictive result. Related modeling techniques and advances developed during the course of the research are described in Chapters 3 and 5.

2.1.3 Methods for Confirming the Functionalities Developed

The last step was to verify the capabilities of the model using one available real oil data set, as possible, and simulated submerged oil data; optimize programming procedures; and disseminate the software and results by means of related publications in specialized journals and conference proceedings. The need for synthetic data for verification was identified, due to the limited nature of available sunken oil data recorded

after a spill. Thus, the model results were not verified versus real data sets measured in time as needed. Partial real data was used for probation if the capabilities imposed by development. This partial input was interpreted from a graphical presentation of spatially-defined qualitative data on oil spill concentrations on the bottom of the Gulf of Mexico following the DBL 152 spill (Barker, 2009). In addition, synthetic oil spill data hypothetically collected in two successive sampling campaigns for a double patch of sunken oil in a nearshore area were generated statistically, and used to verify aspects of model functionality including superposition, limited boundary effects, and multiple sampling times.

Use of Available Qualitative Sunken Oil Concentration Data

Data available for use in this research project was obtained from NOAA for the DBL 152 spill, which occurred on 11 November 2005 in the Gulf of Mexico. The data were available only in the graphical form shown in Figure 2.2. Data were collected by dragging absorbent mop-like samplers, termed *pom-poms*, along the bottom, and recording the geographical trace of the pom-pom and the visually-estimated amount of oil collected as $\leq 1\%$, 5-10%, 11-50%, 51-100%. This figure was interpreted by assigning each recorded relative concentration in percent to the midpoint of the drag trace shown in the figure. Coordinates of the spill location, also shown, and sampling points were approximated based on the figure. These data location and concentration is the result of the transport process that had been occurring between the time of accumulation of the oil on the bottom after the spill and the sampling campaign time; therefore, the information

gathered and input to the model is considered to hold intrinsically Gaussian transport parameters like the instantaneous relative location of a mean with respect to an initial location, a coefficient of diffusion, and variability inherited from the environmental conditions of the ocean at the sampled location and time. The date of this sampling campaign was taken as 25 November 2005, as indicated in the figure. No further information was available, and therefore this information was taken as an example test case for the use of limited information in a spill scenario.

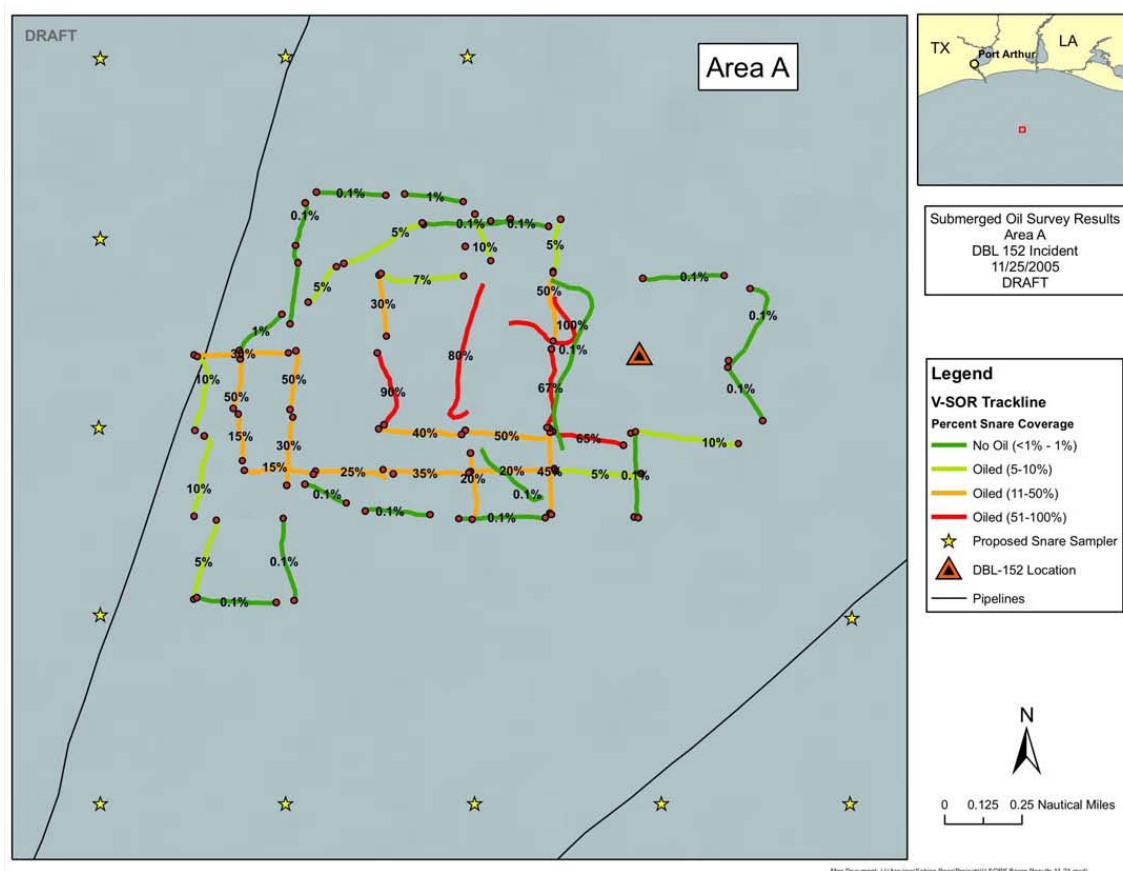


Figure 2.2. Aerial plot presenting the results of a sampling campaign following the 2005 DBL 152 spill near Port Arthur, TX.

Development of Synthetic Spatially-Defined Sunken Oil Concentration Data

To verify additional functionality of the model including superposition, boundary effects, and multiple sampling times, synthetic data was generated. This was done by assuming a roughly bi-modal Gaussian distribution of oil on the bottom, that is, oil in patches (that might in turn have internal patchy character). However, real oil spill data will not be neatly Gaussian in shape and will come from a distribution of experimental error. Therefore, the bi-modal, bivariate Gaussian distribution was used as the assumed mean oil concentration on the bottom, with relative concentration then sampled from a distribution of variability around that mean, exponential in form. The exponential distribution is the most likely distribution of sampling error around a fixed mean, given that concentrations cannot be negative, by the Principle of Maximum Entropy (Jaynes, 1957), as will be described more fully in Chapter 3 in reference to model development. Thus, the concentration, C_i , at each assumed sampling point in space and time was sampled from an exponential error distribution with scale parameter λ , and mean $1/\lambda$ equal to a superimposed bivariate Gaussian distribution as follows:

- Mean concentration at the i -th of 90 (in space) x 2 (in time) samples was found by superposition of two assumed Gaussian patches of sunken oil, as $\bar{C}_i = \sum_{j=1}^2 \gamma_j \mathcal{N}_j$, in which $\mathcal{N}_j = f_j(\mathbf{X}, t | \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, \rho_j)$ is the j -th assumed patch of sunken oil and γ_j is the relative mass of oil contained in that patch. The oil was assumed to occur in a nearshore environment;

- To account for real experimental and natural variability, measured sample concentrations, C_i , were then sampled from an exponential distribution with parameter $\lambda = 1/\bar{C}_i$.

2.2 Development of Methods for Accounting for Approximately Curved Boundary Conditions

A method of computing Lagrangian relative concentration profiles subject to curved, continuous or discontinuous boundary conditions based on the method of images employed in groundwater hydraulics and water quality modeling was developed. The methodology incorporated two phases:

Considering the state of the art about applicable approaches to modeling boundary conditions in two dimensions subject to variable geometry, it was necessary to create a new bi-dimensional technique for calculating any continuous curved boundary or any set of discontinuous curved boundaries with the method of images. A curved boundary results from a continuous number of line segments in any direction (a polyline), that are superimposed to the shoreline. Similar to the internal boundary condition of a finite difference model, the shorter the lines, the better the precision of the method. The graphical dissimilarity here is that the curved boundary is approximated by a continuous polyline, not by a staircase-like set of vertical and horizontal steps. Although not applied in the Sunken Oil Simulator (SOSim), the procedure can also be used for solving multiple, discontinuous bi-dimensional boundaries with the method of images, that is, by

using more than one polyline, or broken sets of line segments approximated to curves. The method requires the idealization of the shoreline, wherewith the application of Fickian transport law can remain pertinent. The developed approach was applied to SOSim to calculate the approximate initial effect of the vicinity of coastlines in oil spill scenarios. The complete method is presented in Chapter 4 as a contribution to be used in other applications.

Developing an understanding of the multi-order or repetitive nature of a mirror image process, and express it by means of an iterative algorithm. During a reflection process, a source is reflected with respect to a boundary and, the produced imaginary reflection is mirrored back with respect to all available boundaries, in a way that the source and the imaginary sides are constantly experimenting imbalances that will occur until the total original mass is equaled at both sides. The new approach is capable of accounting for multiple-order-image superposition with the aim to guarantee sunken oil mass conservation within the bay at all times and modeling scenarios.

2.3 Development of Algorithms for Integration over Highly Dimensional Uncertain Parameter Space

The computational approach that allowed a highly parameterized predictive Bayesian multimodal Gaussian model of sunken oil location was developed using combinatorial math methods across multidimensional parameters for stochastic function-solving and for maximum-likelihood estimation to identify the most likely oil patch(es). The need for

combinatorial math lays in the fact of multidimensionality. Each of the 23 parameters of the model could contain a variable, unknown number of elements. The number of elements within each parameter's domain was fixed so the number of possible discretization values in each dimension was always the same and the number of total Riemann integration differentials was known. This approach allowed for the simulation of operations between every element of the parameter space as if the parameters and their dimensions were arrayed in matrix form. That is, for each possible subset of parameters of the domain, one possible target of every participating function is calculated and fed to the mathematical model, for every mode (or oil patch), and always in two dimensions. The superposition of modes happens repeating the same combinatorial method, but among combinations of previous results (see Chapter 5). Consecutively, the most likely oil patch(es) is identified using the maximum likelihood estimator method. The final integration is performed using a Riemann approach adapted to the combinatorial nature of the process. The complete methodology would not have been possible without the use of programming techniques and computer machines, as described in Chapter 5.

2.4 Development of the Computer Application for the Model

The software development was possible thanks to the Python programming language (Python Software Foundation, 2010) and derived modules like PyQt, PyQGIS, Numpy and Matplotlib, among others. The principal method employed was programming oriented to objects, in which different *classes* represent different algorithms of the model;

within each class, functions called *methods* give the functionality for, and operate, the different capabilities of the model.

The Python code provides the Bayesian model methods for: (1) graphically communicate with the user, through an object-oriented programmed graphical user interface and executable application; (2) calibration of instantaneous oil spill scenarios given limited available field data on oil concentrations in time and space; (3) obtaining sets of printable, georeferenced maps of relative probability profiles at different prediction times of a calibrated scenario; (4) altering or changing both the user-selected geographic areas (open sea or nearshore) and the desired output resolution subject to a predetermined scale and land proximity; and (5) performing post-processing tasks proper of a basic GIS-like software.

Chapter 3. Predictive Bayesian Data-Derived Gaussian Model of Sunken Oil Mass

3.1 Introductory Remarks

Sunken oil is difficult to locate because remote sensing techniques cannot as yet provide views of sunken oil over large areas. Moreover, the oil may re-suspend and sink with changes in salinity, sediment load, and temperature, making deterministic fate models difficult to deploy and calibrate when even the presence of sunken oil is difficult to assess. For these reasons, together with the expense of field data collection, there is a need for a statistical technique integrating limited data collection with transport modeling.

Predictive Bayesian modeling techniques described in the next section have been developed and demonstrated for exploiting limited information for decision support in many applications. The Bayesian modeling approach allows integration of available numerical data together with any less-quantitative information, with rigorous accounting for uncertainty in accordance with the laws of probability. Predictive Bayesian methods involve the development of unconditional probability distributions for the quantity of interest, by integrating over all possible values of the uncertain parameters.

The purpose of this chapter is to describe the development of a predictive Bayesian multi-modal 2-D Gaussian model for computing unconditional relative probabilities of sunken oil location as a function of time based on limited available spatial field data on relative, subjective sunken oil concentrations. Methods are developed to implement

integration over a highly dimensional uncertain parameter space. The method of images is extended for use with ideal curvilinear shoreline boundaries. An interactive graphical user interface is described. Model capability for locating sunken oil patches based on limited field data is confirmed versus data on the DBL-152 spill (Barker, 2009) and capability for projection in time is shown using simulated field data.

3.2 Predictive Bayesian Gaussian Approach

Predictive Bayesian distributions are distributions of unconditional (on knowledge of the parameter vector) probability. The predictive distribution is obtained by multiplying the sampling distribution (e.g., the multinormal), by a Bayesian posterior distribution describing numerical and other knowledge of the parameter vector, and integrating the product over the uncertain parameter space according to the Theorem of Total Probability. The result is a distribution that evolves in shape, becoming progressively narrower as the level of available information increases, converging on the underlying distribution of variability. Figure 3.1 shows this evolution in a general, one-dimensional case.

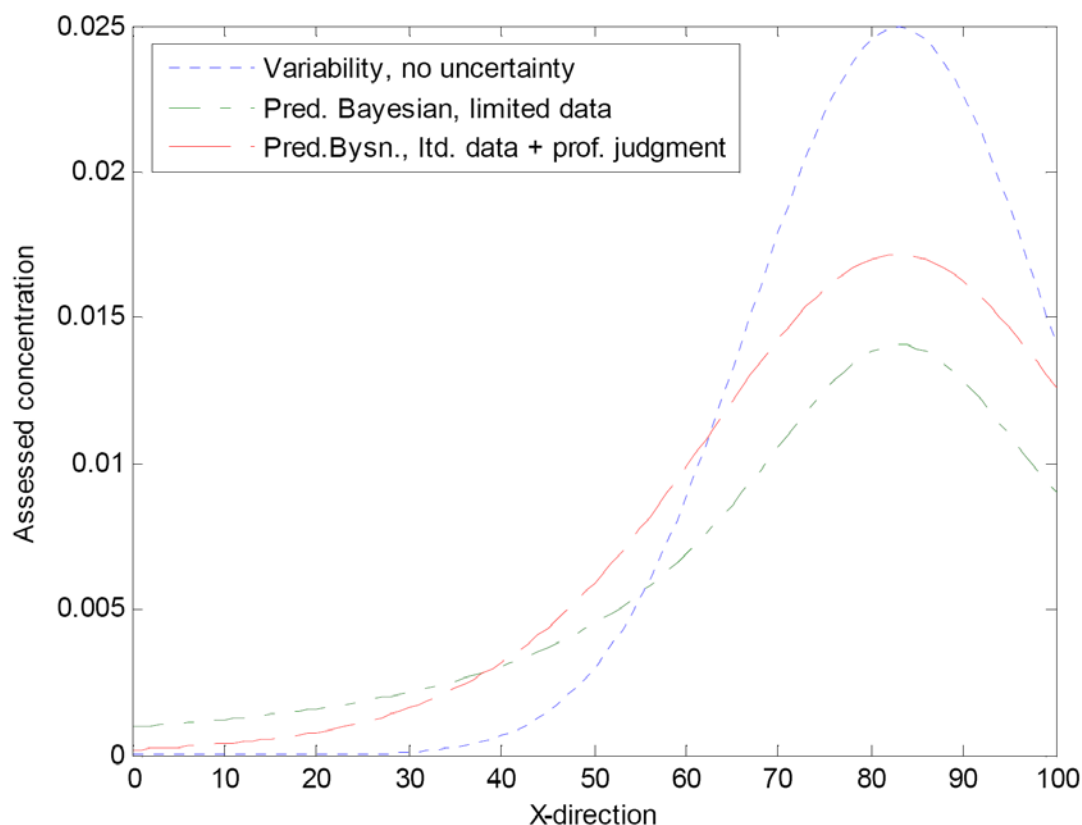


Figure 3.1. Distribution of physical variability assuming perfect information, or no uncertainty, and predictive Bayesian distributions based on limited data, and limited data with professional judgment, showing the decrease in information entropy with increased information availability.

An unconditional probability distribution represents the probability of future outcomes based on experimental samples and refined prior information on its parameters. When prior information is either diffuse or constant relative to the sample information, it is said that the ‘refined information’ or posterior distribution, is solely proportional to the experimental samples (Winkler, 2003). Following this logic, the general predictive Bayesian analytical model that represents the unconditional probability of a particle of oil

at $\mathbf{X}=(\mathbf{x},\mathbf{y})$ is the following expression, involving a conditional sampling distribution and a data-derived likelihood function, in which the integration is over an uncertain parameter space:

$$f(x, y, t) = \int \left(\left(\sum_{j=1}^J \gamma_j f_j(x, y, t | \mu_{x,j}, \mu_{y,j}, \sigma_{x,j}, \sigma_{y,j}, \rho_j) \right) \right) \partial \boldsymbol{\mu} \partial \boldsymbol{\sigma} \partial \boldsymbol{\rho} \partial \boldsymbol{\gamma} \times L(\mathbf{C}_i | \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\rho}, \boldsymbol{\gamma}) \quad (3.1)$$

in which $f_j(\mathbf{X}, t | \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, \rho_j)$ is the j -th Gaussian patch given knowledge of the range of parameters mean $\boldsymbol{\mu}_j = (\boldsymbol{\mu}_x \boldsymbol{\mu}_y)_j$, standard deviation $\boldsymbol{\sigma}_j = (\boldsymbol{\sigma}_x \boldsymbol{\sigma}_y)_j$, correlation coefficient ρ_j , and sunken oil fraction γ_j , with $\sum \gamma_j = 1$ to maintain conservation of mass; $\boldsymbol{\mu}_j = \mathbf{v}_j t$; \mathbf{v}_j is the average advective velocity vector (L/T) of the j -th Gaussian patch; t is time (T); $\sigma_j^2 = 2Dt$ is the standard deviation or measure of the effective “breadth” of the patch at time t ; D is the horizontal average sunken oil coefficient of diffusion (L²/T); and $L(\mathbf{C}_i | \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\rho}, \boldsymbol{\gamma})$ is the likelihood function of the observed concentration data, \mathbf{C}_i , at the irregular sampling locations and times, in which \mathbf{C}_i represents the vector of relative concentration data, $(C_1, C_2, \dots, C_i, \dots, C_I)$, at locations (x_i, y_i) and times, t_i . The parameter space is described as follows:

$$\begin{aligned}
\boldsymbol{\mu} &= \{(\mu_x, \mu_y)_1, \dots, (\mu_x, \mu_y)_J\} \\
\boldsymbol{\sigma} &= \{(\sigma_x, \sigma_y)_1, \dots, (\sigma_x, \sigma_y)_J\} \\
\rho &= \{\rho_1, \dots, \rho_J\} \\
\gamma &= \{\gamma_1, \dots, \gamma_J\}
\end{aligned} \tag{3.2}$$

Note that one of the patches has no γ parameter since $\gamma_4 = 1 - (\gamma_1 + \gamma_2 + \gamma_3)$.

In essence, field data are sampled from a distribution of sampling variability around a mean concentration. This mean is modeled by the value of the multimodal 2-D Gaussian distribution at that point in space. The distribution of sampling variability was assumed to be exponential in form, because the maximum Shannon entropy distribution around a fixed mean over a non-negative range is exponential (Shannon, 1948). This form is proposed as the most likely distribution of concentrations that might be observed at a point on a bay bottom, by the Principle of Maximum Entropy (Jaynes, 1957). That is, entropy is the average log of the inverse height of the density and so, assuming normalization, is a measure of distribution “breadth.” Therefore, maximizing entropy means maximizing the range of feasible outcomes and, all-the-more-so, the number of ways to satisfy the constraints. The maximum- entropy distribution is then realized because it can be obtained in overwhelmingly many more ways. For example, the Gaussian diffusion equation itself results if the entropy of a distribution of diffused mass is maximized subject to independent constraints in terms of the mean (or advective velocity, controlled by pressure gradient) and the variance (or coefficient of diffusion,

controlled by e.g. viscosity, temperature). Likewise, entropy can be the starting point in deriving many physical laws, if the governing constraints are known (Kapur, 1989; Jaynes, 1957).

Assuming an exponential distribution of oil concentration sampling variability around the mean, modeled by the multimodal Gaussian distribution, the likelihood function in Equation (3.1) can be written in terms of the relative concentration as follows:

$$L(\mathbf{C}_i | \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\rho}, \boldsymbol{\gamma}) = \prod_{i=1}^I \lambda_i \exp(-\lambda_i \mathbf{C}_i), \quad (3.3)$$

in which $1/\lambda_i = \sum_{j=1}^J \gamma_j f_j(x_i, y_i, t_i | \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, \boldsymbol{\rho}_j)$. That is, it is assumed that the concentration sampled at a point in space and time comes from an exponential distribution of sampling error with mean $1/\lambda_i$.

The conceptual model, though Bayesian in character to address uncertainty and speed of computation was developed primarily to accept field (sampling) data, that is, to be data-driven only. Due to its structure, the capability for accepting prior information such as derived from bathymetry or professional judgment regarding model parameter values can be added as appropriate.

3.2.1 Conditional bivariate Gaussian distribution

Mass concentration in a fluid subject to advection and dispersion in any media can be shown, for example by maximizing the Shannon entropy of a distribution of (oil)

particles subject to a mean location and variance, to have a Gaussian concentration profile subject to bounds with real mean and real positive variance domains (Shannon, 1948). Conditional on knowledge of the parameter vector, a bivariate Gaussian is adopted as the conditional sampling distribution for the two-dimensional Bayesian model of sunken oil locations on a relatively flat bay bottom. Therefore, we have that:

$$f_j(\mathbf{X}, t | \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, \rho_j) = \mathcal{N}(\mathbf{X}, t | \theta_j) = \frac{1}{2\pi\sigma_{x,j}\sigma_{y,j}\sqrt{1-\rho_j^2}} \exp\left[\frac{-Bm_j}{2(1-\rho_j^2)}\right], \quad (3.4)$$

$$\text{in which } Bm = \frac{(x_m - x_0 - \mu_x)^2}{\sigma_x^2} - \frac{2\rho(x_m - x_0 - \mu_x)(y_m - y_0 - \mu_y)}{\sigma_x\sigma_y} + \frac{(y_m - y_0 - \mu_y)^2}{\sigma_y^2}$$

and $\boldsymbol{\mu}_j$ and $\boldsymbol{\sigma}_j$ represent the two dimensional means and covariance matrices, respectively, for each patch. In Equation (3.4), x_m, y_m are the coordinates of a point to be modeled, and θ_j represents the set of parameters $\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j$ and ρ as described in the conceptual model.

3.2.2 Bayesian posterior distribution

A posterior probability distribution can be used in predictive Bayesian theory to foretell the behavior of another possible observation without the need of performing further experimental sampling on the zone of possible observation.

Bayesian methods involve first assigning a joint (that is, multivariate, to address all uncertain parameters) *prior* probability distribution to the uncertain parameters of a

sampling distribution such as the Gaussian. This distribution is assigned based on non-numerical forms of information, such as bathymetry that might affect sunken oil locations, as function of the unknown parameters. In the model developed, no prior information was assumed, such that the probability prior to data collection of finding oil on the bottom was constant spatially (for relatively flat ocean bottoms, dredged bays, reef flats and lagoons or pools protected by offshore rocks). Therefore, based on the Principle of Maximum entropy, the prior distribution was taken to be a uniform distribution over the uncertain parameter space. This can be done by setting the prior equal to unity as an arbitrary constant, so that it drops out of the equation, because normalization if required is obtained in a subsequent mathematical step.

A Bayesian *posterior* distribution is obtained as a refinement (narrowing) of the prior distribution using Bayes Law:

$$f(\theta | x_1, x_2, \dots, x_I) = \frac{L(x_1, x_2, \dots, x_I | \theta) \varepsilon(\theta)}{g(x_1, x_2, \dots, x_I)}, \quad (3.5)$$

in which $L(x_1, x_2, \dots, x_I | \theta)$ is the likelihood function, $\varepsilon(\theta)$ is the prior, and

$g(x_1, x_2, \dots, x_I)$ is the probability that the data was observed, a normalizing constant with respect to θ , the parameter space. I is the total number of sampling points i . The total mass of sunken oil is generally unknown, and as a result the final predictive distribution need not be normalized, as a distribution of relative, not absolute, concentrations.

Therefore, the normalizing constant, $g(x_1, x_2, \dots, x_I)$, is also not needed, and the (un-

normalized) posterior for the model becomes $f(\theta | x_1, x_2, \dots, x_I) \propto L(x_1, x_2, \dots, x_I | \theta) \mathcal{E}(\theta)$
 $= L(x_1, x_2, \dots, x_I | \theta)$. That is, $f(\theta | x_1, x_2, \dots, x_I) = \prod_{i=1}^I \lambda_i \exp(-\lambda_i C_i)$, where

$$1/\lambda_i = \sum_{j=1}^J \gamma_j \frac{1}{2\pi\sigma_{x,j}\sigma_{y,j}\sqrt{1-\rho_j^2}} \exp\left[\frac{-Bi_j}{2(1-\rho_j^2)}\right]_j, \quad (3.6)$$

$$\text{and } Bi = \frac{(x_i - x_0 - \mu_x)^2}{\sigma_x^2} - \frac{2\rho(x_i - x_0 - \mu_x)(y_i - y_0 - \mu_y)}{\sigma_x\sigma_y} + \frac{(y_i - y_0 - \mu_y)^2}{\sigma_y^2}$$

The intrinsic physical information contained in the collected field data along with the overall information about the spilled oil, the accident location, and a set of elapsed times are used to calibrate a spill scenario, that is, to obtain a posterior probability distribution, which in this case is proportional to the likelihood function as shown.

3.2.3 Multimodality and Superposition

The model was developed as a multimodal Gaussian, to account for example for the presence of multiple patches of oil collecting on the bottom, or for oil concentrating more highly in deep bathymetric features at any or all times. The predictive model is able to infer from the data whether or not the sunken oil is distributed in single or multiple patches, and to track and predict this multimodal behavior in time. As a consequence, the assessed concentration profile can be, or become, either multimodal or unimodal in time. Results depend upon the data, as well as on boundary conditions, desired prediction times, and the resolution selected by the user.

The total mass of sunken oil is not typically known as a function of time, considering ongoing sinking and re-suspension processes. Therefore, computations of the model arbitrarily assume a constant mass of sunken oil with time, producing maps of relative sunken oil in space which identify areas with the most oil. As such, relative masses cannot be compared across time. Therefore, in an emergency response scenario, the occurrence of increases or decreases in the total mass of sunken oil over time would need to be assessed from the field data, for example, by inspection. With this basic assumption, the model can infer the relative mass of oil contained in each patch from the field data. This relative mass is represented by the parameter γ_j , representing the fraction of total sunken oil that belongs to the j -th patch, which can be considered the mass weighting parameter among patches. Summation of the γ_j to a constant value over time was therefore one important test of internal model consistency used for verifying the model as it was developed.

An expansion of the Bayesian Bivariate Gaussian analytical expression for multiple modes shows how superposition is mathematically attained in the model:

$$f(\mathbf{X}, t | \theta)_{overall} = \sum \gamma_j \mathcal{N}(\mathbf{X}, t | \theta_j), \quad \sum \gamma_j = 1. \quad (3.7)$$

That is, the overall conditional Gaussian sampling distribution is composed of the sum of any possible combinations of a weighting parameter, the fraction γ_j , among patches and a bivariate normal distribution \mathcal{N}_j with parameters $\mu_{x_j}, \mu_{y_j}, \sigma_{x_j}, \sigma_{y_j}$ and

ρ_j , that represents a single mode or oil patch j , such that the sum of the weighting parameters is equal to one. To maintain this sum equal to unity over multi-dimensional integration operations with some computational efficiency, an algorithm based on combinatorial mathematics was developed, as described in the section Algorithm and Code Development.

3.3 Assumed or Specified Parameter Domains

An unconditional probability distribution represents the probability of future outcomes based on experimental samples and refined prior information on its parameters. When prior information on the parameters is either diffuse or constant relative to the sample information, it is said that the ‘refined information’ or posterior distribution, is solely proportional to the experimental samples (Winkler, 2003). This being the case, the model was developed primarily to be data-driven, as opposed to driven by prior information on the parameters.

As such, the model was designed with capability for operation without specific information on (or distributions of) average advective velocity vectors and diffusion coefficients found on the bay bottom. Other explanations for such a design of the model include that: (1) empirically-derived relationships between transport parameters and the scale of a contaminant plume such as those developed by Okubo (Okubo, 1971), have been observed in the surface of the ocean; but (2) studies on the propagation or parallelization of surface phenomena to the bottom of the ocean have concluded that only

near a 1% of the total energy observed at the surface is replicated at the bottom, and is observed especially during storm events (Beegle-Krause *et al.*, 2006), and therefore (3) experiments conducted on the surface do not apply and are unknown to occur at the bottom. As well, literature suggests that (4) average bottom current conditions will not be sufficient to move the oil in a continuous manner (Beegle-Krause *et al.*, 2006; Boehm *et al.*, 1981). Rather, with enough energy, tarmats in the bottom generally break up into smaller pieces of oil that spread out into a larger area, similarly to sediments in terms of episodes of burial and re-exposure and mobilization. Lastly, (5) the energy required to move the oil on the bottom varies with depth and site and is unknown. A break-up energy level has been estimated as $6 \text{ m}^2/\text{Hz}$ (Beegle-Krause, *et al.*, 2006).

For all the previous reasons together, prior information on the physical parameters was not readily available to the model, and is considered 'diffuse' or constant, tending to one in obedience to Bayesian inference methods (Winkler, 2003).

. Accordingly, default ranges of values were assigned for each of these parameters with the mere purpose of evaluating and integrating equation (3.1) over a wide set of physically-possible values. The default ranges are no more than assigned, changeable possible values with restricted domain (e.g. a coefficient of diffusion cannot be zero or negative in time); they are not expected values of the parameters. Currently, default ranges are wide, as shown in Table 3.1, having been assigned based on literature and professionally-judged values together with a margin of uncertainty. In the future, these ranges can be refined and can be based on the input of prior information. As of now, the

graphic user interface gives the user the possibility of changing input in the form of ranges of parameter values based either on hydrodynamic data or professional judgment, to over-ride default ranges assigned in the research. The default ranges are shown in Table 3.1.

Table 3.1. Default parameter ranges of the model.

Parameter	Symbol	Units	Minimum	Maximum
Velocity in the east-to-west direction	v_x	km/d	-3.0	3.0
Velocity in the north-to-south direction	v_y	km/d	-3.0	3.0
Coefficient of diffusion, east-to-west direction	D_x	km^2/d	0.01	0.89
Coefficient of diffusion, north-to-south direction	D_y	km^2/d	0.01	0.89
Coefficient of correlation between directions	ρ	[-]	-0.999	0.999
Mass conservation weighting parameter	γ	[-]	0 (fixed)	1 (fixed)

Displacement in the west or south direction from the initial point of bottom contamination implies that the velocity vector has a negative direction. Displacement in the east or north direction from the point of initial collection implies that the velocity vector has a positive direction. As the ranges of the parameters are not expected values and do not represent prior information on the parameters, the probability of the sunken oil transporting eastward or westward are set balanced (-3 to 3 km/d). The same applies for the southward and north velocities.

As observed, the default domain of the coefficients of diffusion is restricted to positive, non-zero values. The physical domain of non-default ranges, assigned by the user, is restricted by the graphic user interface of the model.

The default range of the coefficient of correlation is defined based on its typical domain in a bivariate Gaussian distribution.

The domain of the mass conservation weighting parameter is fixed. Since the total sunken oil mass is unknown, this range must (a) vary between and (b) be normalized with respect to, two known constants; in this case, between zero and the unit.

The resulting prediction is not obtained through maximization of the likelihood function; rather, it is obtained after integration under the complete parameter space, a combinatorial method (Chapter 5) that does not allow tracing back something such a “most likely” set of source parameters. A relative probability is assigned to every user-defined coordinate (modeling node) at the given user-defined prediction time by means of a single integration over the entire parameter space per coordinate. If no probability is found at any of the selected geographic locations (modeling nodes), no result will be shown for the area, which means that Equation (3.1) has the potential to not finding a solution if the parameter space is not suited to adjust probabilities of finding future states of the sampled data within a given zone. The last is considered prove that the model does have the capability of accounting for the physical parameters and does not select a solution based on the maximization of the likelihood function. Since the default ranges of

integration over the parameter space are all physically possible, equifinality should be mathematically attained between the resulting prediction obtained through integration over the physically-possible unknown parameter space and another simulation that used another method to actually find the parameters that such simulation described. The fundamental problem of proving equifinality would be that the highly dimensional parameter space does not allow for the obtaining of a closed analytical maximizable expression that can be solved for the 23 parameters in question. Even random-walk or chain processes to find maximum likelihood estimators of the parameters would need a complex numerical computation.

3.4 Projecting the Oil Mass in Time

The concentration profile in a (one dimensional) media following the instantaneous introduction of a mass, M , can be found either starting with the Fick's Law-based advection-diffusion equation and applying the inverse Fourier transform (Chin, 2006), or solving the random walk model for a particle (of oil) (Einstein, 1925). The result takes the form of a Gaussian distribution with variance growing in time:

$$c(x,t) = \frac{M}{A\sqrt{4D_x t}} \exp\left(-\frac{(x_0 - vt)^2}{4D_x t}\right), \quad (3.8)$$

in which D_x is the uncertain diffusion coefficient of a Gaussian path in one dimension [L^2/T], vt is the distance the mass M has traveled from the initial point x , v is the

uncertain mean velocity vector of the Gaussian patch [L/T], and A is the transverse area of mass introduction. In the model developed, the mass per unit area, M/A , of sunken oil is taken as an arbitrary constant.

Comparing the fundamental solution of the diffusion equation (Equation 3.7) to the equation of a Gaussian distribution (Equation 3.4 in one dimension), it is seen that $\mu = vt$ and $\sigma = \sqrt{2Dt}$. Furthermore, by the time oil mass reaches the bottom of the sea from the initial aerial spill location (x_0, y_0) , it will already be characterized by an initial velocity and an initial diffusion coefficient, which results in:

$$\boldsymbol{\mu} = \mathbf{x}_0 + \mathbf{v}t \quad \text{and} \quad \boldsymbol{\sigma} = \boldsymbol{\sigma}_0 + \sqrt{2\mathbf{D}t} . \quad (3.9)$$

Equations 3.8 are parameterizations of uncertain parameters that are used along the Gaussian model which varies as a function of user-defined times of oil mass prediction.

The Gaussian nature of the sampling probability function and its parameterization in time ensures that no parameter will hold physically impossible values in time, for example, the standard deviation of a predicted Gaussian patch at a given time will always hold values smaller than those held by future states in time because the prediction time factor, t , is a known, unique coefficient within any single integration-solving process.

3.4.1 Incorporating Multiple Sampling Times

In order to account for changes in sunken oil movement due to e.g. tidal action, storm events, and sediment entrainment, the model was built with capability for accepting data from multiple sampling campaigns (sampling campaigns performed at different times and different areas) that would lead to updated calibrations of oil spill scenarios. Input of data subsequent to the initial sampling campaign increases the number of product terms in the likelihood function, increasing the information content of the calibration and increasing information concerning oil movement, in particular.

The likelihood function of the observed data, C_i , at each i -th irregular sampling location and times with respect to the uncertain parameter space is:

$$L(C_i | \mu, \sigma, \rho, \gamma) = \prod_{i=1}^I \lambda \exp(-\lambda C_i) \quad (3.3)$$

in which C_i represents the vector $(C_1, C_2, \dots, C_i, \dots, C_I)$ of measured concentrations at different location-times. The SOSim GUI allows up to ten different sampling campaigns or times, with an unrestricted number of sampling locations at each time.

3.4.2 Taking Into Account the Potential for Sinking and the Short-Term Weathering

The model infers oil location as a function of time by comparing the initial point of accumulation with data on its distribution at a later point in time. If only one sampling campaign is available, the initial point of accumulation is critical to these predictions in

time. While many times the spill location and time are known, and may represent the initial point of accumulation for floating oil, the initial time and point of accumulation are not as well known for sunken oil, depending on oil density, sediment entrainment, and other factors. In fact, while oil may take a week or more to start a weathering process that may result in sinking, Type 4, 5 and 6 oils start a process of sinking by gravity immediately after abrupt release or during a continuous spill event (each oil type at its own pace).

For the SOSim model, the initial spill time is indicative of the major release at the actual coordinates of the accident at the surface, but the approximated time at which oil starts to collect at the bottom is really what defines the framework of the model with respect to time. The difference between the spill time and the time at which oil starts to collect at the bottom is considered a “retardation factor”, included in the model as a function of the oil type as identified by the user. Oil is identified as one of the accepted Types 1 through 6 described in the literature review, accounting for multiple properties of oils related to the potential for sinking by gravity and for weathering by evaporation and/or dissolution. Accordingly, all times that modify the likelihood of finding sunken oil at a given location are adjusted by the *retardation factor*, RF , as $t = t - RF$, in which RF reflects an assessed, generalized delay associated with the particular oil type before starting to collect at the bottom. It is assumed that no sampling campaigns will be useful if performed before the assessed retardation gap.

3.4.3 Development of Curved Boundary Capabilities by the Method of Images

An extension of the method of images (or method of image charges in electromagnetics, or reflection theory in physics) was used to approximate the overall initial mass retention effect of curvilinear coastal boundaries when the sunken oil is predicted to be transported near a coastal area. The physical reality of the ocean boundaries (the shoreline) is approximated to an ideal condition in which a “relatively-flat ocean bottom” is truncated by a theoretically vertical barrier that is assumed to retain the sunken oil without compromising the oil’s mass conservation and the Fickian nature of the transport process; that is, it is assumed that phenomena that might occur close to a shoreline environment such as oil adsorption onto the barrier’s material or flow into the porous medium do not occur. Figure 3.2 replicates the physical approximation in one dimension. The method, developed in Chapter 4, is used as a numerical approach to calculate the effect of the vicinity of coastlines in oil spill scenarios.

The scheme used in this research also involves the assumption of uniqueness of the current modeling conditions, maintaining conservation of mass, while redistributing it consistent with each source and at each time within the nearby boundaries imposed by the geometry of the coastal zone, approximated and projected to the bottom (see Figure 3.2). A novel implementation using a polyline approximation of shoreline geometry was developed and executed in the model, as described in Chapter 4.

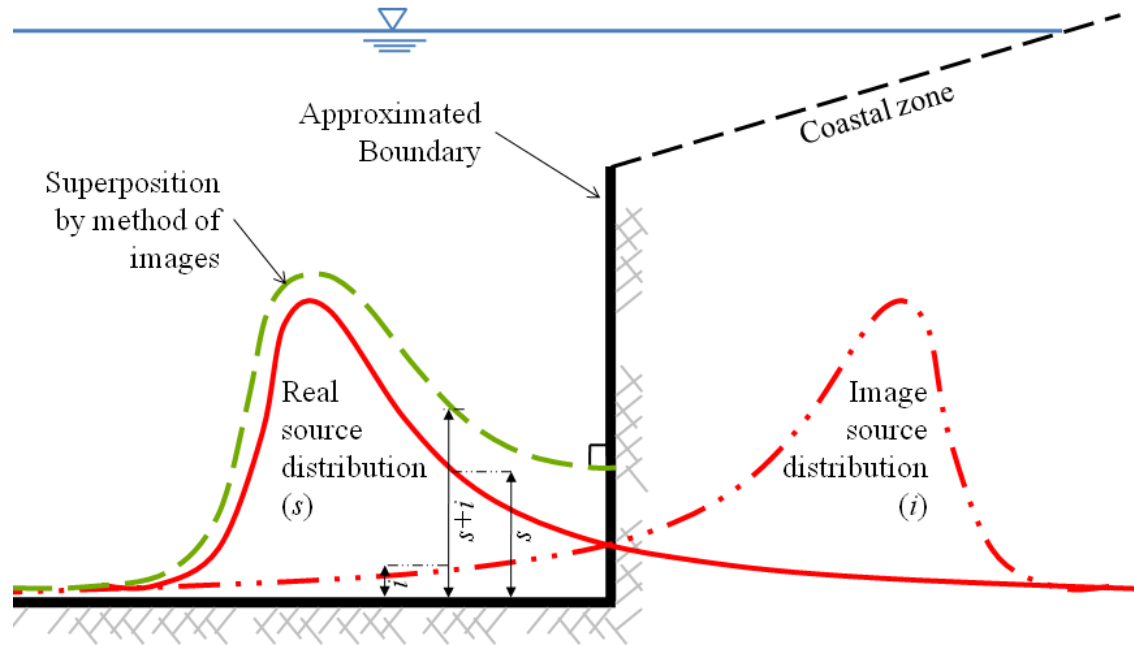


Figure 3.2. Graphical depiction of the physical idealization of the boundary conditions to apply the Method of Images in the SOSim model.

Consider an example in which an oil patch is denominated ‘the source’. The source occurs at a certain point $(x_0, y_0) = (x, 0)$ on the bottom of a bay and has a concentration distribution $c(x, y)$. This point lies at some distance x from the shoreline which, by definition, has a concentration gradient of zero. As the patch approaches the shore, oil at the leading edge begins landfall. It is then assumed that this oil mass is reflected back into the water, producing a retention in the nearshore environment by the principle of superposition. In one dimension the retention is easily computed as a sum of the inflowing mass and the reflected mass. However, in two dimensions when the shoreline

is not straight, reflections are not perpendicular with respect to only one line and calculations become cumbersome. Therefore, (relative) mass accumulation is computed in SOSim by setting up imaginary conditions to replicate the curve of the boundary. The method of images momentarily replaces the boundary by an “image source”, equal in concentration distribution but located at opposite coordinates $(-x, 0)$ across the boundary. This new situation is equivalent to the original layout because superposition of the source and its image concentration distributions will simulate the zero concentration gradient at the boundary and will account for spreading and contention of oil. The result is then the summation of concentration distributions on the source side, and all results imagined to occur beyond the boundary can be ignored. Figure 3.2 depicts the concept.

In the single boundary case described above, no other boundary has effect on any of the sources, and thus the reflection process stops at once and the case is considered of zero order. When multiple boundaries coexist, imaginary sources are not perpendicular with respect to only one line but instead it is assumed that they are perpendicular with respect to each segment of a boundary interpreted as a series of connected lines, where every single source is reflected with respect to each of the line segments. In the application of the method of images in groundwater hydraulics to modeling flow near aquifer boundaries, the source is reflected with respect to each horizontal or vertical aquifer boundary, and each subsequent reflection is in turn affected by the boundary that did not reflect it before (Bear, 1979, Todd and Mays, 2005). The approach to considering perpendicular or parallel boundaries (that are not necessarily contiguous) used in

groundwater hydraulics, was adapted and implemented for the case of non-orthogonal contiguous line segments representing approximate geographical boundaries to sunken oil movement. That is, the requirement for perpendicular or parallel boundaries was relaxed using a new method developed to address the change in correlation between the x and y directions. The multi-reflection process creates a series of mass imbalances that must be balanced back with further operations. As a result, computation must be iterated to attain both mass conservation and zero-concentration gradient values at the shoreline.

To approximate shorelines with polylines, the user is shown a map of the spill area, and instructed to click on the map to define up to 10 vertices (on a continuous coastline). These vertices are connected by straight lines to approximate the coastal zone contour. The algorithm allows the addition of as many superimposed terms as needed to the conditional Gaussian bivariate distribution. However, computational demands increase nonlinearly with the number of vertices, due to the nature of the numerical integrations across the multi-dimensional parameter space and the combinatorial numerical approach. Therefore, the program SOSim incorporates a limit of ten vertices.

This approach, like many numerical methods, requires a stopping rule to define the limit of iteration needed to obtain an acceptable solution. In this research, it was found that the sum of second and lower order reflections provided sufficient detail to obtain an observable mass balance in the water body. Thus third and higher order reflections are neglected, greatly reducing computational demands (currently model runs require several hours on desktop computers). Also, because the Gaussian tail is mathematically infinite,

reflection is calculated beginning when a patch center reaches a distance of 2σ from the shoreline, and ending when the tails of the outgoing source and the incoming image have already “crossed” (refer to Figure 3.2) and no longer sum to a value higher than the concentration at the mean $\pm 2\sigma$. To prevent calculation of sunken oil projections at times beyond the estimated time of landfall, when the patches would appear to “bounce” off of the shoreline, a warning message is issued by SOSim when the requested prediction time is estimated *a priori* to be beyond the time of landfall. Such prediction times are beyond the reasonable capability of SOSim to assess relative concentrations.

3.5 Integration to Obtain the Predictive Relative Concentration Profile

Several different integration techniques were considered for the multivariate integration expressed in Equation (3.1), which expanded, is the equation solved in the model to obtain unconditional, relative probabilities of sunken oil mass:

$$f(x, y, t) = \int \left(\sum_{j=1}^J \left[\gamma_j \left(\frac{1}{2\pi\sigma_{x,j}\sigma_{y,j}\sqrt{1-\rho_j^2}} \exp \left[\frac{-Bm_j}{2(1-\rho_j^2)} \right] \right) \right] \right) \partial \boldsymbol{\mu} \partial \boldsymbol{\sigma} \partial \rho \partial \gamma \quad (3.10)$$

$$\times \left[\prod_{i=1}^I \lambda \exp(-\lambda C_i) \right]$$

where $\boldsymbol{\mu}_j = \mathbf{v}_j t$; \mathbf{v}_j is the unknown average advective velocity vector (L/T) of the j -th Gaussian patch; t is time (T); $\sigma_j^2 = 2Dt$ is the standard deviation or measure of the effective “breadth” of the patch at time t ; D is the unknown horizontal average sunken oil coefficient of diffusion (L²/T); and $\left[\prod_{i=1}^I \lambda \exp(-\lambda C_i) \right]$ is the likelihood function of

the observed concentration data, C_i , at the irregular sampling locations and times, in which C_i represents the vector of relative concentration data, $(C_1, C_2, \dots, C_i, \dots, C_I)$, at locations (x_i, y_i) and times, t_i . Bm_j is the expression
$$\frac{(x_m - x_0 - \mu_x)^2}{\sigma_x^2} - \frac{2\rho(x_m - x_0 - \mu_x)(y_m - y_0 - \mu_y)}{\sigma_x \sigma_y} + \frac{(y_m - y_0 - \mu_y)^2}{\sigma_y^2}$$
 for the j th sunken oil patch. The unknown parameter space is comprised of v_j, D_j, ρ_j and γ_j , as explained in Table 3.1.

The first integration technique considered to solve equation (3.9) was an analytical solution, which is available for the predictive Bayesian multinormal distribution (Aitchison and Dunsmore, 1975). However, no solution is available for the multimodal analog that would allow the inference of varying numbers and weights of multiple patches of oil. Second, the use of Markov chain Monte Carlo (MCMC) simulation was considered, to generate vectors of random variates sampled from the Bayesian posteriors over which averages could be computed at each point in space and time. MCMC is the most popular approach to computation of Bayesian posteriors. However, the approach was not considered likely to estimate posteriors successfully, due to the high dimensionality of the model, without the development of new computational approaches. In addition, the principal advantage of the approach is the ability to compute the normalizing constant for the likelihood function, not needed in the SOSim model. Moreover, the distributions applicable to the uncertain parameters of a Gaussian distribution are not highly skewed, and are therefore relatively easy to integrate as a

discretized sum. Therefore, it was concluded that the most efficient approach would be a direct Riemann sum approximation. The approach involves approximating the volume under a surface (or area under a curve) as a sum of small differential volumes, partitioned over the domain of the parameter space.

The Riemann sum requires one initial input: the approximate domain of the uncertain parameters. As mentioned, default values for this range are included in the model based on literature information and statistical principles, and handled as explained in the section Software Development. The Riemann sum integration, consumes considerable computational resources in the SOSim model because of model dimensionality and the programming structure required in the Python programming language used. Algorithms are explained under Software Development, including how the resolution of the integration (the number of discretizations of each parameter range) is related nonlinearly to the number of oil patches and to the precision of prediction.

3.6 Verification of the Functionalities of the Model

The capabilities of the model were verified using one real (as possible) and one synthetic data set. First, the ability of the model to locate sunken oil patches from limited available field data on the DBL-152 spill in the Gulf of Mexico, a major spill involving sunken oil, was confirmed. The event occurred 11 November 2005. Oil was observed to collect in a single patch about 28 nautical miles from shore near Port Arthur, TX, 14 days after the spill. In such cases when the spill is not expected to impact the shoreline, as

determined by the user when assigning the modeling geographic scale, shoreline boundary conditions are not taken into account by the model, saving considerable computational time. To then verify model functionality in terms of boundaries, superposition, and multiple sampling campaigns, the developments were tested using synthetic data on sunken oil collecting nearshore, initially in two overlapping patches and then dispersing along divergent paths. In the second scenario, two sampling campaigns are assumed. Shoreline boundaries are accounted for based on a polyline approximation of the shoreline entered interactively by the user using the graphical user interface (refer to Chapter 4 and Appendix A).

Table 3.2 indicates required and optional input and default and optional output for this verification case. The international system (SI) of units (km, days) is used in model calculations, whereas geographical user input is in terms of coordinates expressed and shown on non-projected maps in the World Geodesic System (WGS).

Table 3.2. Input and output for Verification Scenario 1: DBL 152

Input	Output
<p>Required input:</p> <ul style="list-style-type: none"> Spill coordinates and time: 29.205° N, 093.4683° W (United States Coast Guard, 2005), on November 11, 2005 (General Counsel for Natural Resources/NOAA Office of Response and Restoration, 2009). at zero hours (midnight, assumed as the initial hour of most significant loss) 	<p>Default output:</p> <ul style="list-style-type: none"> Maps of relative concentration (unconditional probability) at desired times (Figure 3.6, 3.7 and 3.8) <p>Optional (post-processing) output:</p> <ul style="list-style-type: none"> None for this scenario

Input	Output
<ul style="list-style-type: none"> • Subjective relative concentration data (Figure 3.4 obtained from Figure 3.3) and sampling time from Figure 3.3: November 25, 2005, 14 days after the spill • Desired prediction dates and times: 12 hours after sampling, 17.5 days after the spill, and 19.5 days after the spill <p>Optional input:</p> <ul style="list-style-type: none"> • User-defined modeling area: approximately 0.2° around the spill, total about 0.4° longitude x 0.4° latitude • Interactive spatial scale and resolution: 40 grid nodes in each direction 	

The sampling campaign provided by NOAA (Barker, 2009) shown in Figure 3.3 was interpreted as follows. The relative oil concentration, in percent, recorded for each sample taken was assigned to the midpoint of its graphically-depicted sampling path, through which the pom-pom sampler was apparently dragged prior to observing the amount of oil collected. Thus the distribution of subjective concentrations shown in the shaded area of Figure 3.3 and detailed in Figure 3.4 was assigned as input for SOSim.

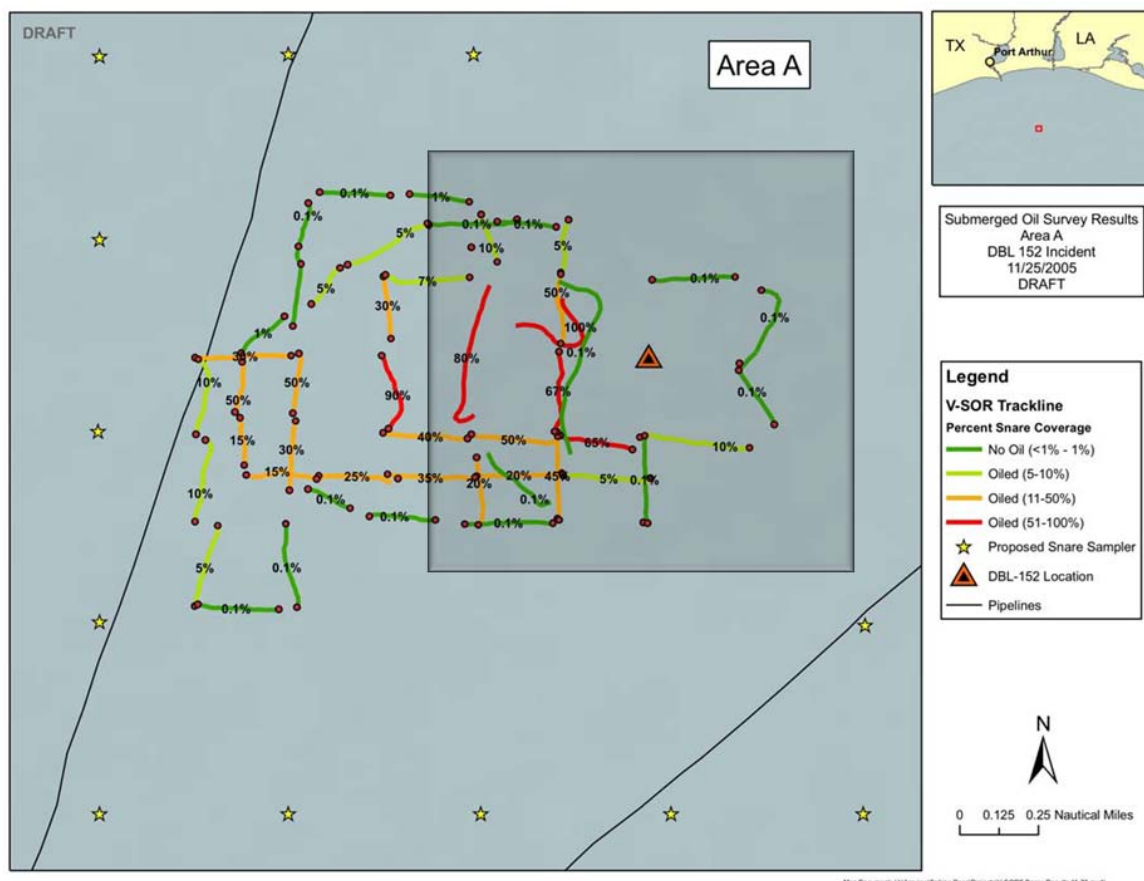


Figure 3.3. Aerial plot presenting the results of a sampling campaign following the 2005 DBL 152 spill near Port Arthur, TX. The shaded area contains the portion of the data used to confirm the ability of the model to locate sunken oil.

The apparent date of data collection is shown in Figure 3.3 (November 25, 14 days after the spill) and the approximate time of most significant loss was assumed as 12:00 AM (midnight). Note that the sampling points in Figure 3.3 are scattered randomly throughout the sampling area as was found expedient during the response effort. Therefore, SOSim was built with capability for accepting data at random points in space, as opposed to data sampled according to a lattice (regular) sampling plan.

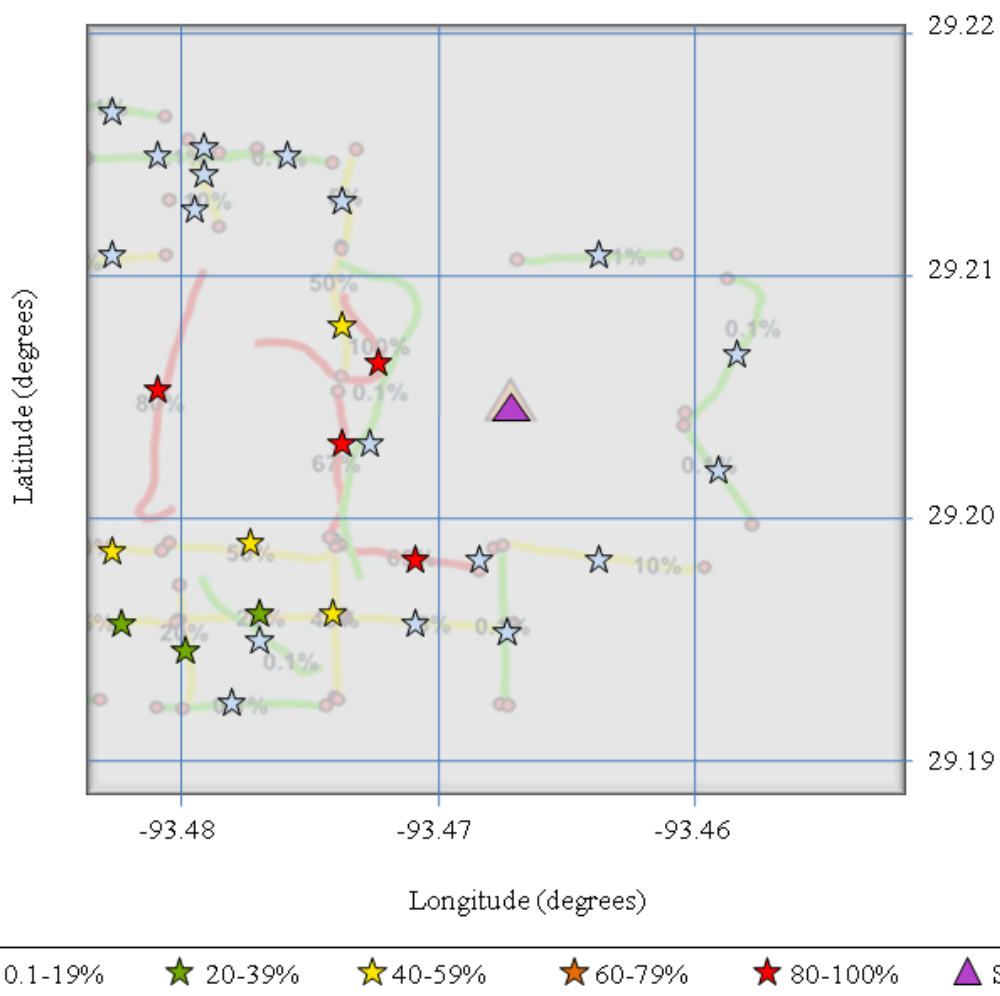


Figure 3.4. Zoom of the shaded area of Figure 3.3, showing the recorded relative concentrations input to SOSim and their geographical location used to confirm the ability of the model to locate sunken oil.



Figure 3.5. General location of the DBL 152 spill at 29.205° N, 093.4683° W.

All input data were entered following prompts of the graphic user interface, as described in the Users Manual (Appendix A). The map of assessed relative unconditional probabilities of finding sunken oil at 12:00 PM (noon) on November 25, 2005, based on half the available data points as shown in Figure 3.4, is shown in Figure 3.6. Maps of such probabilities at 12:00 PM on November 28, 2005 and 12:00 PM on November 30th, 2005, are shown in Figure 3.7 and Figure 3.8. Prediction times correspond to times of 14.5, 17.5, and 19.5 days after the spill.

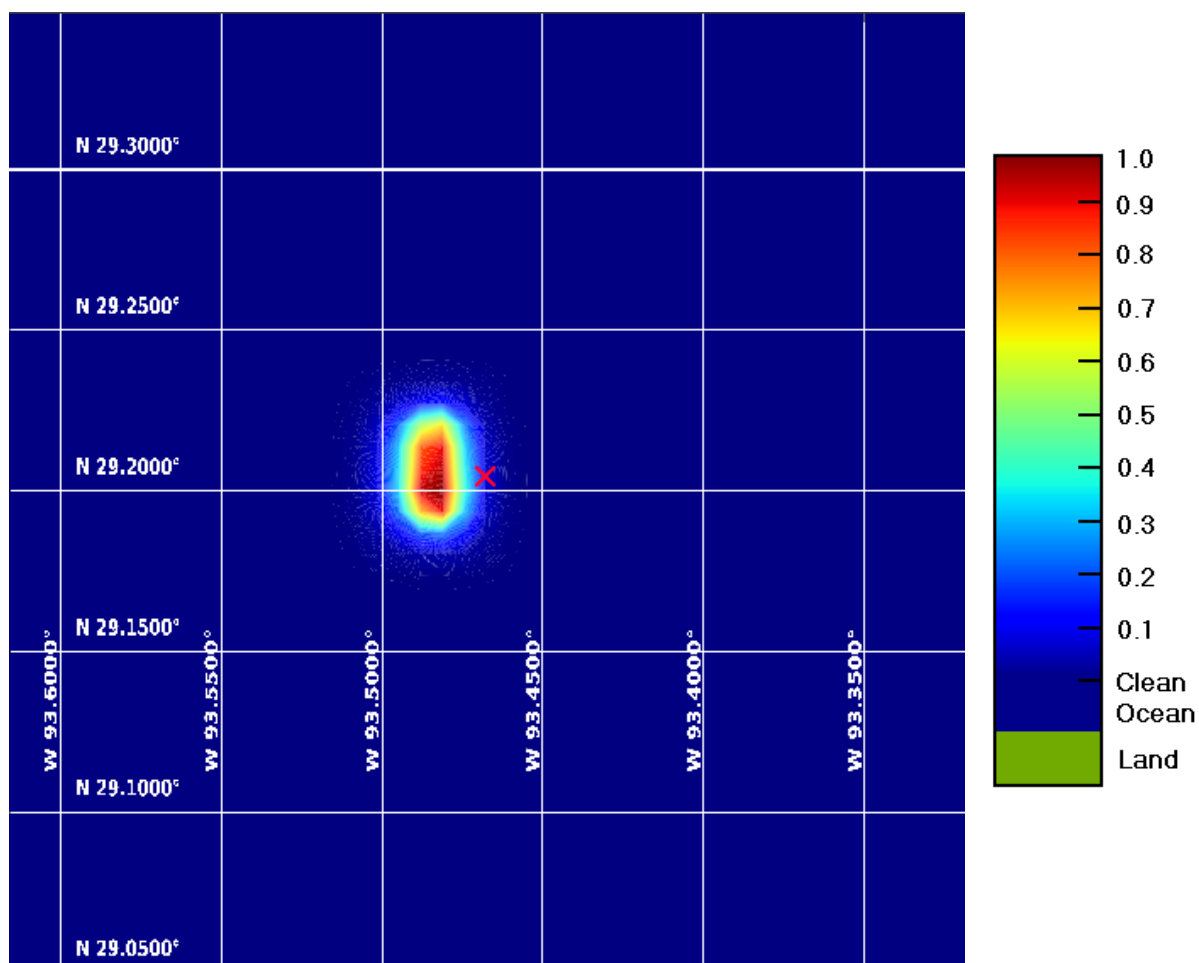


Figure 3.6. Relative probability of finding sunken oil 12 hours after the spill.

Figure 3.6 shows relative probabilities a short time after the first sampling campaign. Given that the location and shape shown in the figure resemble the given data, it is deduced that the model has the capability of interpreting the sampling campaign well. The GUI prevents the user from requesting prediction times prior to the most recent sampling event.

If the user needs a prediction between sampling campaigns, only the sampling campaign(s) conducted prior to the requested prediction time should be entered and on file.

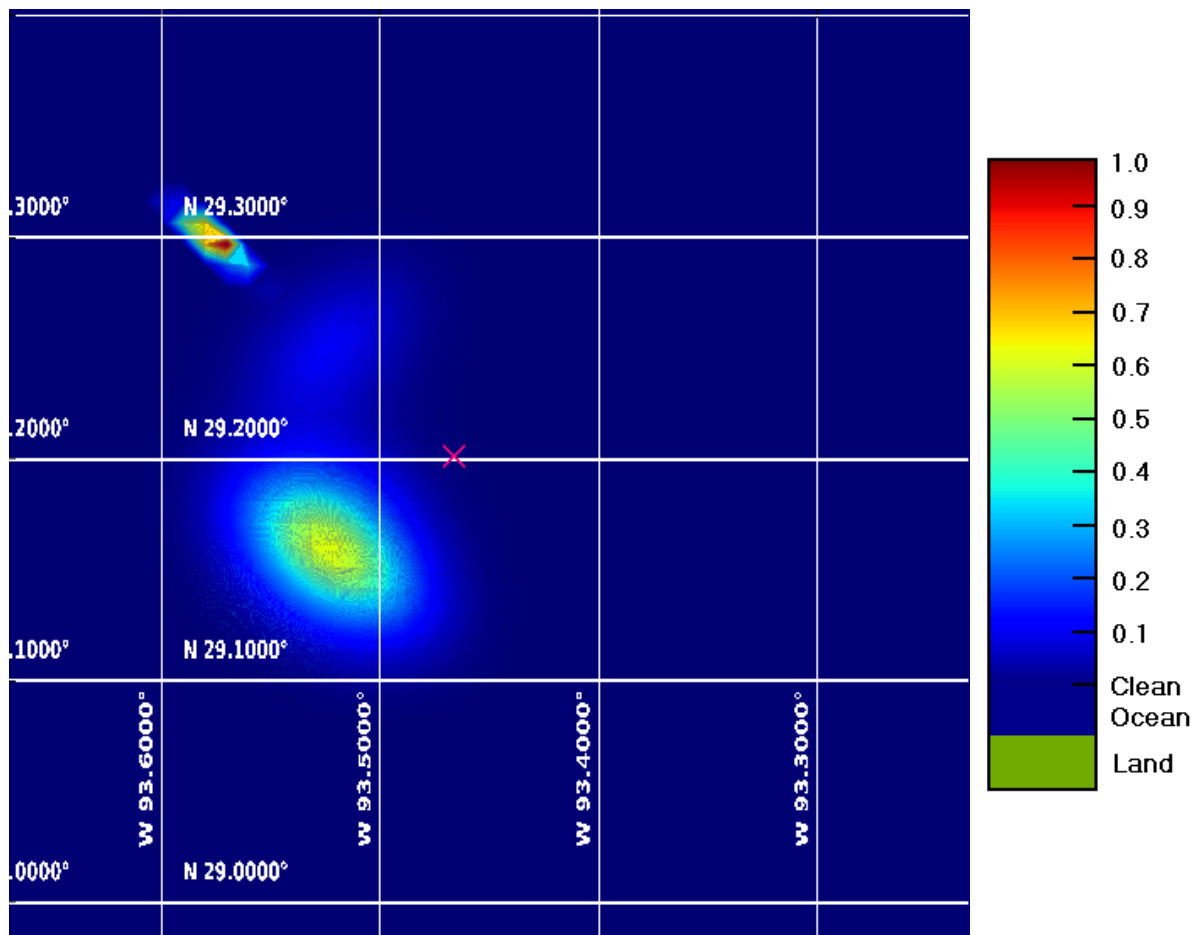


Figure 3.7. Relative probability of finding sunken oil 17.5 days after the spill.

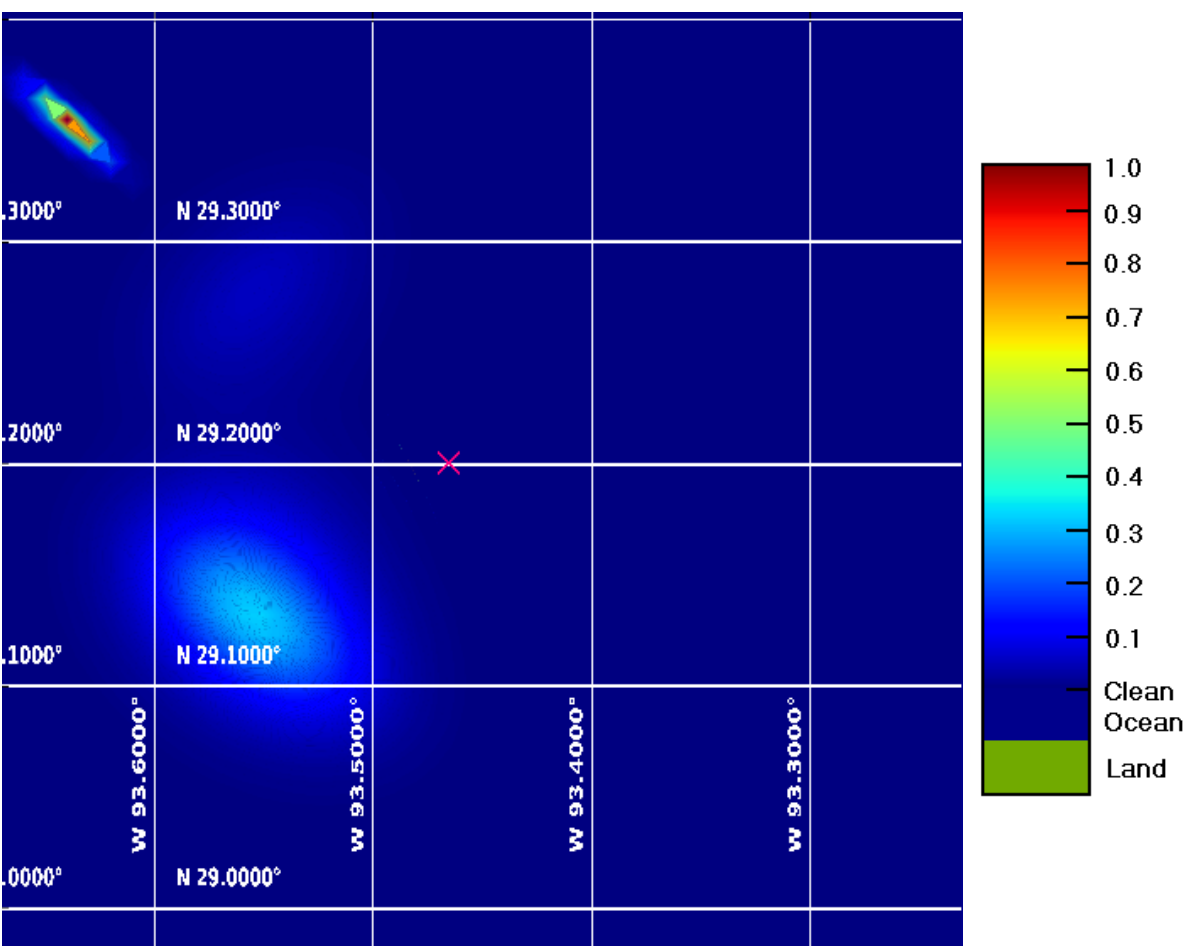


Figure 3.8. Relative probability of finding sunken oil 19.5 days after the spill.

Although red in Figures 3.6-3.8 represents the highest relative unconditional probability of finding oil in each figure, the concentration at such hotspots decreases in time as the sunken oil patches advance in space and disperse. That is, the hue scale is not constant from one figure to another. As shown, the formation of two satellite patches in addition to the main patch was predicted by SOSim by inference based on the data of the

single sampling campaign available and the assumed time of oil collection on the bay bottom. By way of comparison, NOAA Scientifics wrote “We envision the T/B DBL-152 oil breaking up into smaller pieces and dispersing horizontally as it moves downcoast” (Beegle-Krause *et al.*, 2006). This vision is not inconsistent with the figures developed using SOSim, in that the oil breaks into smaller patches and two of them spread while moving offshore and horizontally, with one headed towards land.

3.6.1 Test Scenario 2: Synthetic Multi-Modal, Multiple Sampling Campaign Data on Relatively Flat-Bottom Bay within Coastal Environment

Table 3.3 indicates required and optional input and default and optional output for this confirmation exercise:

Table 3.3. Input and output for Scenario 2.

Input	Output
<p>Required input:</p> <ul style="list-style-type: none"> Assumed spill coordinates and time: (32.80° N, 079.465° W), the 1st of January of 2010. Synthetic subjective relative concentration data and sampling times: Figure 3.9, first sampling campaign at 4 AM on January the 7th,, 2010, made of 90 data points; and second sampling campaign at 4 AM on January the 11th, 2010, also with 90 data points Desired prediction dates and times: 4 AM January the 9th, (two days after the first sampling campaign and 8 days after the 	<p>Default output:</p> <ul style="list-style-type: none"> Maps of relative concentration (unconditional probability) at desired times: Figure 3.11, Figure 3.12 and Figure 3.13. <p>Optional (post-processing) output:</p> <ul style="list-style-type: none"> Revised maps based on subsequent data (new sampled area), Figure 3.12 and Figure 3.13.

Input	Output
<p>spill), 4 AM January the 12th, (one day after the second sampling campaign), and 4 AM, January the 14th, 13 days after the spill.</p> <p>Optional input:</p> <ul style="list-style-type: none"> • User-defined modeling area: about 0.27° latitude x 0.27° longitude within the coastal area • Interactive spatial scale and resolution: default (25 x 25 grid nodes) 	

The purpose of this second synthetic scenario was to test model capabilities in terms of projecting patch locations in time near curved shoreline boundaries using data from multiple sampling campaigns, as a sunken oil patch approaches a coast line. The statistical sampling procedure described in a previous chapter of this dissertation was used to simulate data for the exercise. Figure 3.9 depicts the simulated data and spill coordinates. Figures are rendered for the oceanic area sampled and affected by the initial spill; the general region showing the location of the spill is as in Figure 3.10. Each marker in Figure 3.9 represents one simulated sampling point, having a relative concentration in percent as indicated by the color key below the plot.

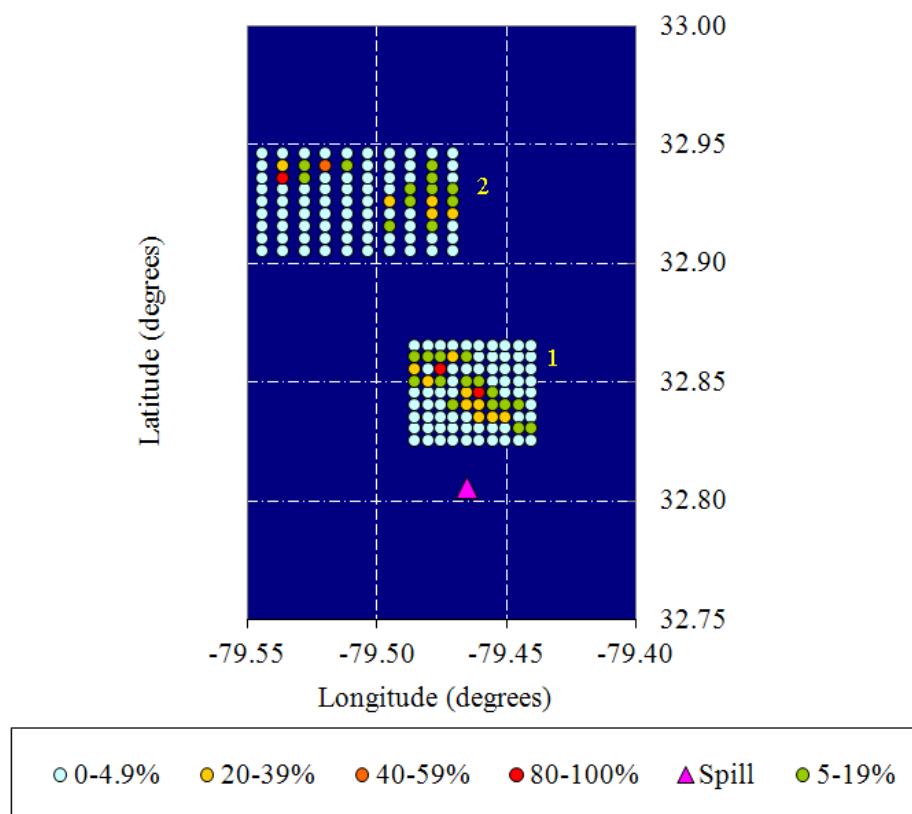


Figure 3.9. Synthetic data on relative sunken oil concentrations in percent, for samples assumed collected on two different days, (1) 6 days after the spill, and (2) 10 days after the spill.

The modeled spill is located at (32.8 °N, 079.465 °W), as shown in the figures. In Figure 3.9, both sampling campaigns 1 and 2 have been synthetically derived to represent two separate patches of oil collecting on the bottom, as indicated by the colors shown. The first sampling event covers an approximate area of 25 km² (9.8 mi²) and the second an approximate area of 35 km² (13.67 mi²).

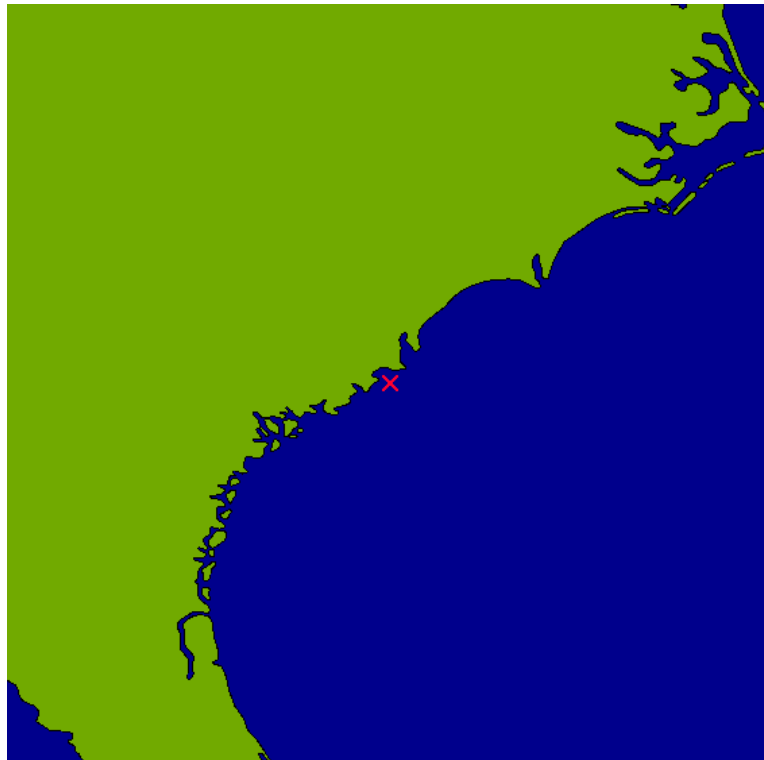


Figure 3.10. General location of the simulated spill scenario of Scenario 2

Figure 3.11 to Figure 3.13 show maps of relative unconditional probabilities of finding sunken oil, interpreted roughly as relative sunken oil concentrations, on the bottom given the imposed boundary conditions, within the user-selected geographical area.

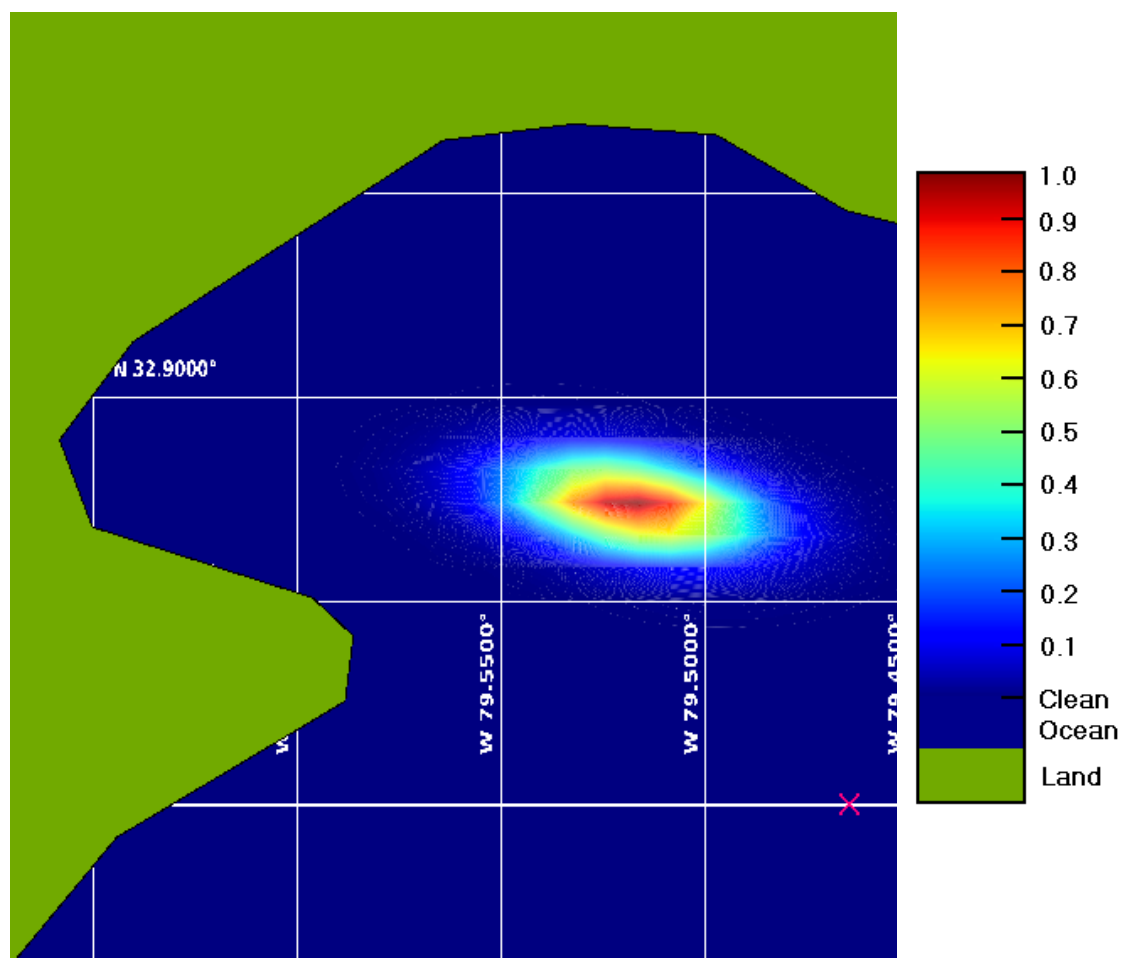


Figure 3.11. Relative probability of finding sunken oil 8 days after the spill (1 day after the first sampling campaign).

Although the first sampling campaign represents two overlapping patches, so close to each other that the model infers a single patch.

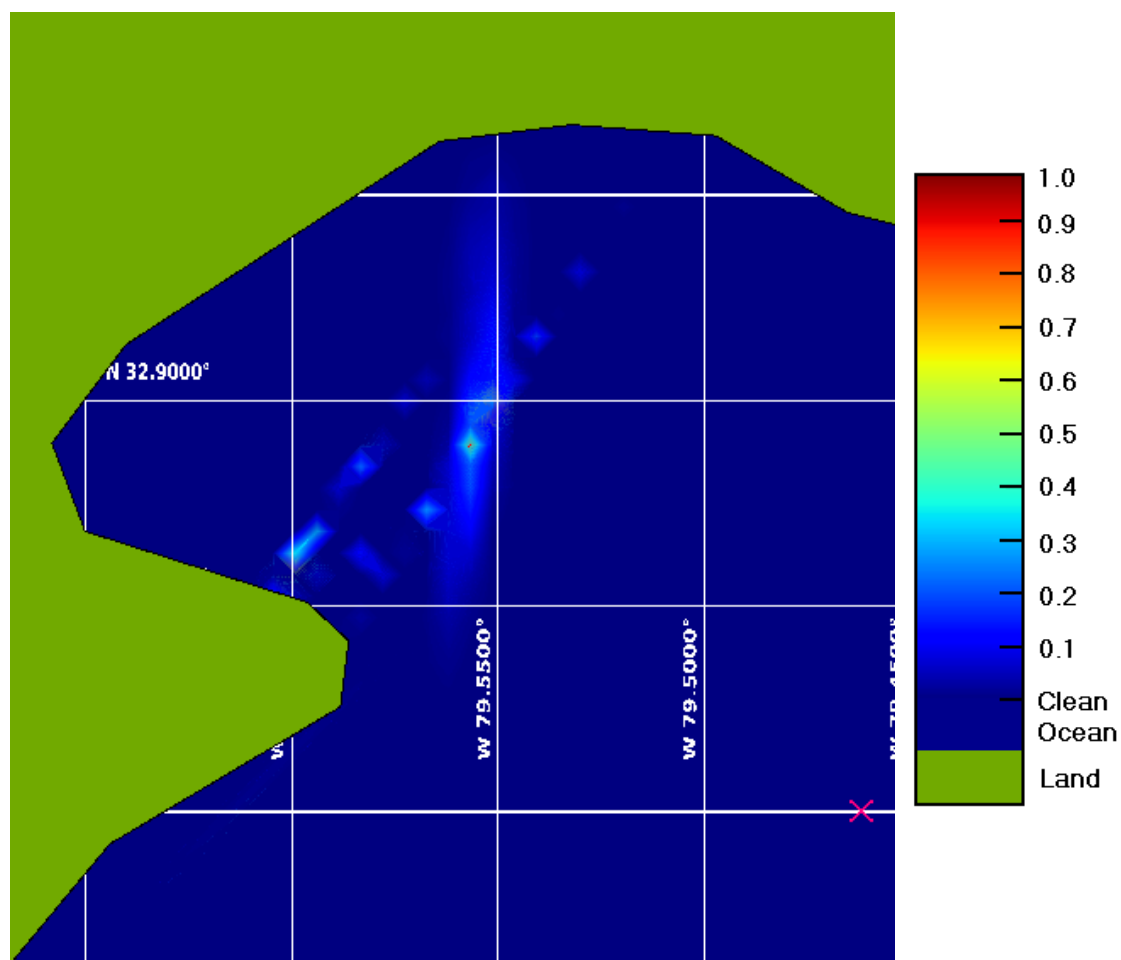


Figure 3.12. Relative probability of finding sunken oil 11 days after the spill (5 days after the first sampling campaign and 1 day after the second sampling campaign), updated based on the second data set.

Figure 3.12 represents an updated prediction considering the second sampling campaign. After 11 days the oil mass is dispersed and is not predicted to impact the shoreline significantly.

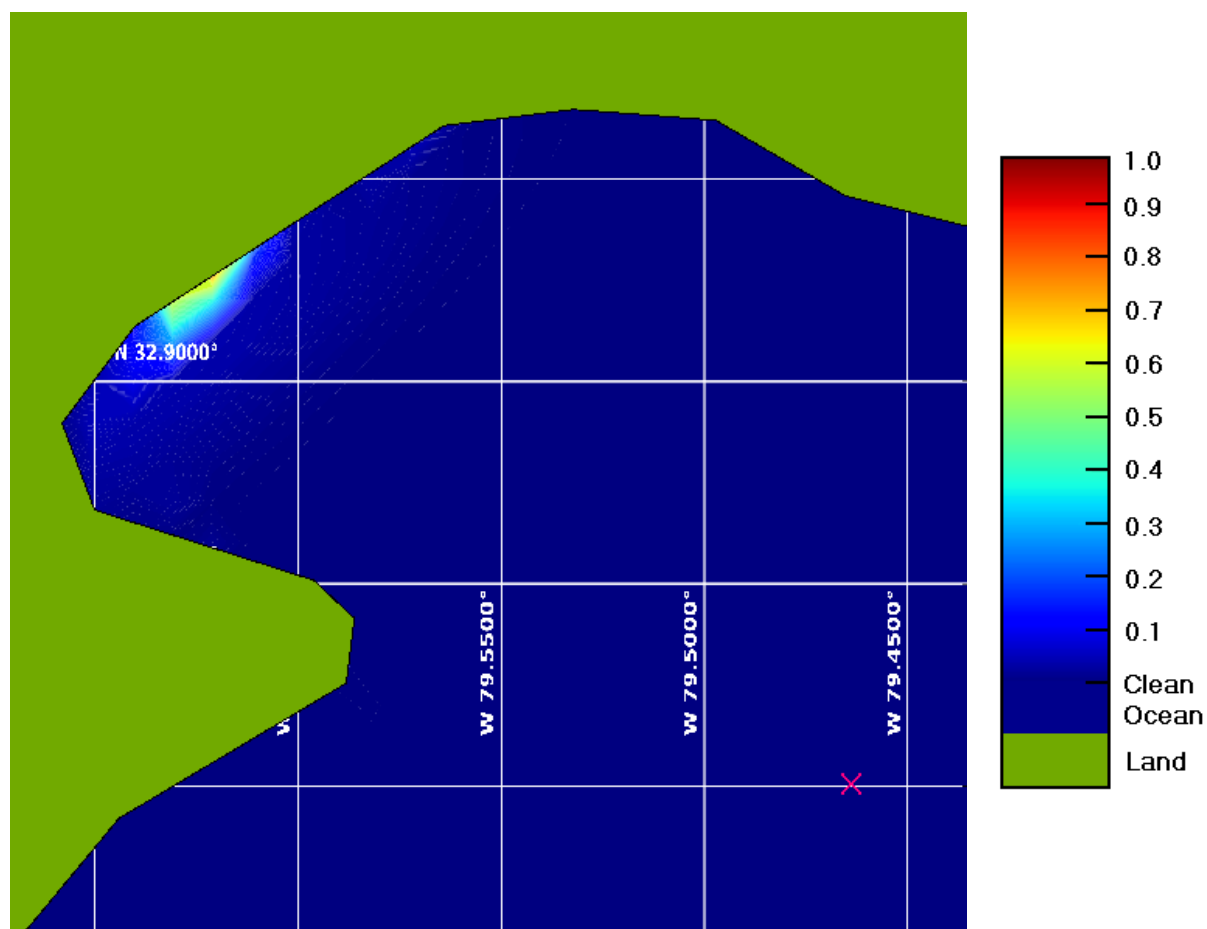


Figure 3.13. Relative probability of finding sunken oil 13 days after the spill (3 days after the second sampling campaign).

After 13 days of the assumed oil spill and using an updated calibration file resulting from one synthetic offshore sampling campaign and one synthetic nearshore sampling campaign, the sunken oil is predicted to impact an ideal barrier close to the coastal zone and to be retained as a result, as shown in the 2-D output map of Figure 3.13. In subsequent versions of the model, the effect of bathymetry, such as slope away from the

shoreline, can be considered as prior information to refine these results. Predicting oil locations following resuspension of the oil by storm events would generally require data to be collected after the storm. Overall, results observed are considered conservative and representative for a relatively flat-bottomed bay environment, given current model capabilities.

3.7 Assessment of the Physical Parameter Domain

A likelihood function represents the observed values of an experiment as function of the parameter space (Θ). A joint likelihood function is the joint probability of several experimental observations as function of Θ . Irregularly-sampled points that belong to one (or different) sunken oil concentration sampling campaign(s) are the observed values. In predictive Bayesian inference, the parameter space is unknown although possible parameter ranges of integration have been utilized merely with the aim to restrain the limits of the stochastic integration, as described in previous sections. The integration is performed using a combinatorial algorithm (explained in Chapter 5) that samples parameter values from the restrained domain in an organized and uniformly stepped manner to calculate the necessary distributions including the superimposed joint likelihood function. Many invalid combinations (such as those in which superposed oil patches (j) did not conserve the original mass by $\sum_{j=1}^J \gamma_j \neq 1$ or those with superposition values close to the computational precision limits) are discarded, while many others are factored in the likelihood function. An analysis of the joint likelihood of

the observations obtained as function of the possible and valid combinations of parameters of the domain allows for the assessment of the physical relevance of the selected integration range. The exercise has been performed using the observations of the limited sampling campaign of the BDL-152 oil spill shown in Figure 3.3, for which predictions have been mapped in the previous section. Figure 3.14 has been obtained for the valid combinations of the parameter domain (θ_k), where K is the total number of non-discarded parameter combinations.

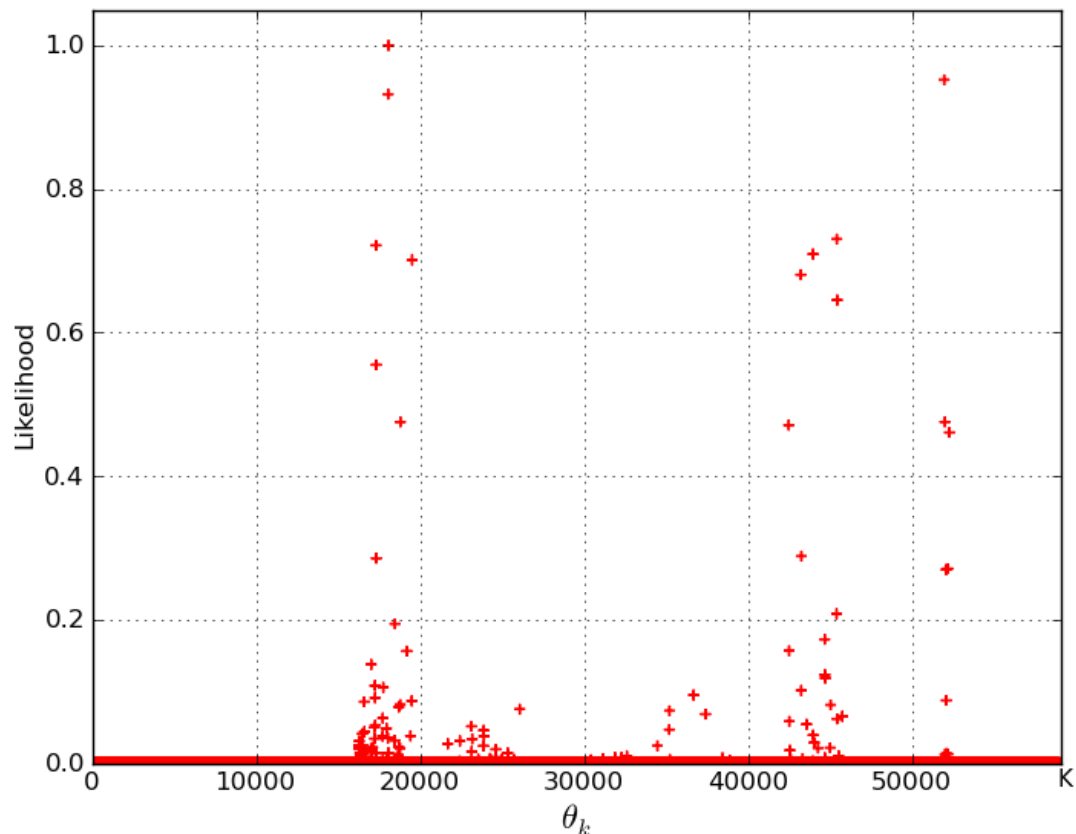


Figure 3.14. Assessment of the likelihood of the observed limited data of the DBL-152 case as function of valid parameter combinations (θ_k) within the parameter space (Θ).

It is deduced from Figure 3.14 that there is one valid combination of parameters θ_k that maximizes the likelihood of the observations that have been input to the simulator. This likelihood analysis plot also shows that only a few possible parameter sets are driving the model. Using a data mining programming technique over the combinatorial algorithm, it was possible to trace back the parameter set (θ_k) that produced such maximum; they are shown in Table 3.4.

Table 3.4. Parameter set that produces the maximum likelihood of being observed given the parameter space (Θ) for the limited data obtained of the DBL-152 oil spill.

Parameter			Sunken oil patch j			
			1	2	3	4
Advective velocity	x	(km/d)	-1.5	-1.5	-2.25	1.5
		(ft/s)	-0.057	-0.057	-0.085	0.057
	y	(km/d)	1.5	1.5	-1.5	-1.5
		(ft/s)	0.057	0.057	-0.057	-0.057
Coefficient of Diffusion	x	(km^2/d)	0.67	0.67	0.89	0.89
		(ft^2/s)	83.47	83.47	110.88	110.88
	y	(km^2/d)	0.89	0.56	0.45	0.56
		(ft^2/s)	110.88	69.76	56.06	69.76
Coefficient of correlation ρ_j		[-]	-0.24975	-0.4995	0.999	0.24975
Weighting parameter γ_j		[-]	0.625	0.25	0.125	0

The parameter set (θ_k) indicates that the data is inferred to be distributed in three sunken oil patches, each carrying 62.5%, 25% and 12.5% of the total mass. The fourth patch and its parameters are discarded since the sunken oil weighting parameter $\gamma_4 = 0$ (contains no oil mass). From the values, it is concluded that the default parameter range for the advective velocity is adequate but the upper boundary of the default range of the coefficient of diffusion in the x (E-W) direction could be extended since the current maximum ($0.89 \text{ km}^2/d$) is repeatedly taken as a sign of the possible need of a larger maximum value (see Table 3.1 to review the default possible ranges of the parameters). Maximum-likelihood parameters on the boundary of the range imply that the physics of the problem might be likely to occur at values that are out of the proposed range. In this case, the oil patch may be dispersing at a higher rate. The Gaussian coefficient of correlation has a standard domain that needs not to be changed. The same applies to the weighting parameter, which represents the total oil mass to be conserved at all times distributed among patches.

Other than the default parameter range, its discretization can also be analyzed using the values in Table 3.4. Peaks or modes are repeatedly observed along patches, from which it is assessed that a denser selection of parameters around those values is required to attain optimum precision. However, there are currently only 9 (uniformly distributed) discretization values per parameter range; with enhanced computational processing power, the number of discretizations can be increased, which would imply an increment

of the resolution of the Bayesian inference and the improvement of the precision of the parameter values that maximize the likelihood (Table 3.4).

Lastly, Gaussian distributions using the parameter values of Table 3.4 can be plotted for each patch using the parameterizations $\mu = x_0 + vt$ and $\sigma = \sigma_0 + \sqrt{2Dt}$, and assuming $x_0 = (0, 0)$, $\sigma_0 = 0$ (unrealistic) and $t = 1$, to theoretically compare with the experimental data set of DBL 152. Figure 3.15 shows the resulting plot.

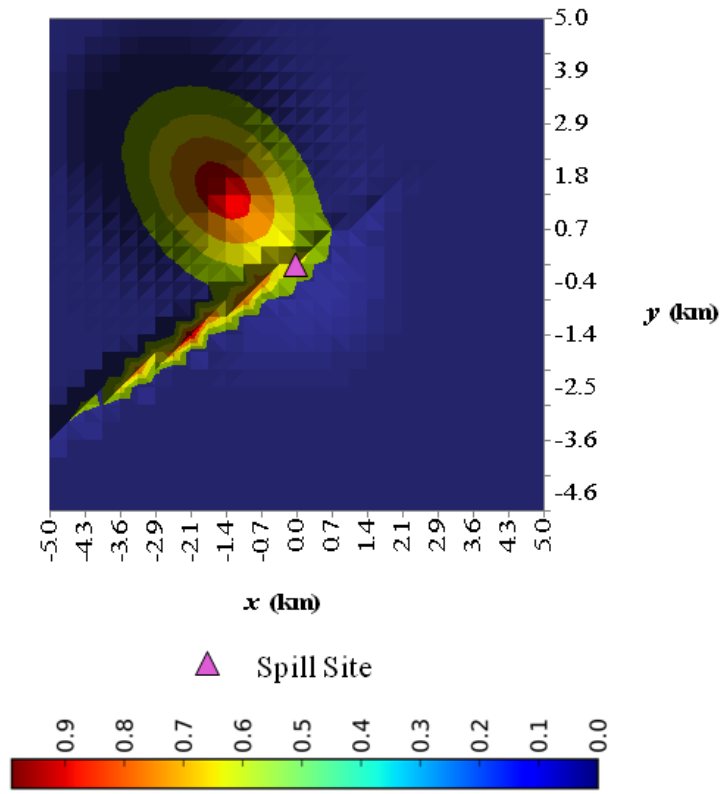


Figure 3.15. Graphical representation of the superimposed Gaussian distributions that maximize the likelihood function obtained by SOSim for the DBL 152 limited data campaign.

A subjective interpretation of Figure 3.15 would state that there is a possible match with the sampled data shown in Figure 3.3: the first and second patches overlap and approximate to one patch that embodies the data points located to the northwest and around the spill site, the third patch shows a distribution with a prolonged tail in the southwest direction which can be interpreted to account for the rest of the data.

The analysis of the parameters that would maximize the likelihood of observing the DBL 152 data set suggests equifinality. However, the assessment of the physical parameter domain, a default input of the Sunken Oil Simulator SOSim, would lead to stronger conclusions if performed for various real data sets not available at this point.

3.8 Discussion and Importance of Results

As described in this chapter, the model was shown able to utilize irregularly-sampled, limited available field data on relative occurrences or concentrations collected from a relatively flat bay bottom shortly after a spill event to predict locations of sunken oil. Sampled data intrinsically includes, related to the spill location and time, information on the physical forces that drive the transport process, identified as Fickian in nature. The model also offers capability for updating of results based on additional, relative data that modifies the calibration files (understood as the likelihood function), from possibly different and irregular geographical areas, as they become available. The model is able to present unconditional probabilities of sunken oil in output maps at user-requested times of prediction. The model accounts for the time between spill and time of deposition on

the bottom that oil may experience due to its potential for sinking and short-term weathering. The software has also proven multimodality capability, that is the aptitude of the predictive model to infer from the data whether or not the sunken oil is distributed in single or multiple patches, and to track and predict this multimodal behavior in time. Finally, the model has the functionality of inferring oil location in time based on subsequent synthetic data together with the approximation of the initial effect of boundaries. However, SOSim is not intended for modeling the incidence of sunken oil mats on the shore, but to simulate sunken oil from offshore spills including the approximate retention effect imposed by an idealized coastal zone. Also, much further confirmation of temporal capability is recommended.

The proposed predictive Bayesian multi-modal Gaussian model, SOSim Version 1.0rc1, is operational for use with limited, randomly-sampled, available subjective and numeric data on sunken oil concentrations in relatively flat-bottomed bays. Model operation and the approach in different spill modalities (for example, for continuous-discharge, abrupt bathymetry or deep water rig accidents), however, is recommended for continued development.

The SOSim model represents a new approach, coupling a Lagrangian modeling technique with predictive Bayesian capability for computing unconditional probabilities of mass as a function of space and time. The resulting prediction is not obtained through maximization of a likelihood function (although equifinality is suggested by a maximum likelihood assessment of the input parameter range); rather, it is obtained after integration

under the complete parameter space, a combinatorial method for computing unconditional multimodal probability distributions (details in Chapter 5). The result is an unconditional relative probability distribution that is wider than the posterior distribution resulting from the sampled data, because it now contains a degree of uncertainty.

The physical mechanisms driving the new distribution are the same that had been able to develop the physical state and distribution of the sampled data set from the time of the spill until the sampling campaign time. The approach can be applied to continuously update calibrations so that predicted states reflect the most current physical changes and intrinsic uncertainties and variability induced by the environment on the measured data. Incorporated are new methods for computation of two-dimensional boundary conditions by images (details in Chapter 4), and combinatorial techniques for computing unconditional multimodal probability distributions (details in Chapter 5). The approach addresses the current need of NOAA to rapidly deploy modeling capability without readily accessible information on ocean bottom currents. Another aspect is the stand-alone application developed, including a user-friendly interactive graphical user interface that can be employed without the need for knowledge of Bayesian or other modeling techniques, to track sunken oil during emergency response missions.

Chapter 4. Computing Lagrangian Relative Concentration Profiles Subject to Curved, Continuous or Discontinuous Reflecting Boundary Conditions

4.1 Overview

Lagrangian Bayesian Gaussian modeling techniques are not readily enabled for complex boundary conditions. Methods to implement such boundary conditions, that could be used along stochastic modeling of sunken oil in a bay, include the consideration of the existing method of images in two dimensions (Todd and Mays, 2005, Bear, 1979), the detailed study of the variable geometry offered by the different modeling scenarios, the creation of numerical procedures to implement variations of the method of images, and the understanding of the mechanisms to guarantee oil mass conservation within the bay at all times and modeling scenarios. However, the implementation is limited to Gaussian-distributed sources.

This chapter explains in detail the method developed for use in the Bayesian Sunken Oil Simulator (SOSim) model to compute relative bi-dimensional Gaussian probability (or relative concentration) profiles in the sea subject to curved, continuous reflecting boundary conditions. In SOSim, the boundaries correspond to the shoreline and the modeling area is the sea. Although SOSim was designed to model scenarios that only include bays and relatively flat-bottomed sea waters; and although scenarios with discontinuous, multiple or small-scale land shapes such as oceanic areas with isles, islets, or straits, are not addressed in SOSim (due to computational demand), the method developed and described in this chapter is theoretically capable of accounting for any

kind of boundaries and modeling areas including discontinuous, multiple and (untested) small-scale geographic features. The approach developed in this research involves the assumption of uniqueness of the current modeling conditions (that is, variability from one modeling scenario to another), maintaining conservation of mass, while redistributing it consistent with each source and at each time within the boundaries imposed by the coastline geometry.

4.2 Methods and Background

The method of images employed in groundwater hydraulics and water quality modeling is the foundation of the approach. The methodology incorporated two phases: (1) the revision of the standard method of images in two dimensions and (2) the creation of single and general case scenarios, along with geometric approximations and engineered computed algorithms that matched such approximations.

4.2.1 Revision of the Standard Method of Images in Two Dimensions Applied to Groundwater Hydraulics

A discharging well at a certain point (x_0, y_0) of an aquifer has a discharge flow rate Q . This point lies at some distance a from an impermeable boundary to the west direction, and some distance b to another impermeable boundary to the south direction. The impermeable boundaries, by definition, have a flow gradient of zero. Figure 4.1 shows the sectional view of the example in the dimension east-west. Figure 4.2 depicts

the situation in two dimensions. In the case of groundwater hydraulics, the discharge through the well affects the steady state of the aquifer, causing a *cone of depression* to form. The cone of depression is modeled using certain equations, which will be compared to a Gaussian distribution in the case of this dissertation. Under the presence of an impermeable boundary, the flow gradient zero is obtained by superimposing the cones of depression of the discharge (source) well and its image's. Figure 4.1 shows the resultant cone of depression.

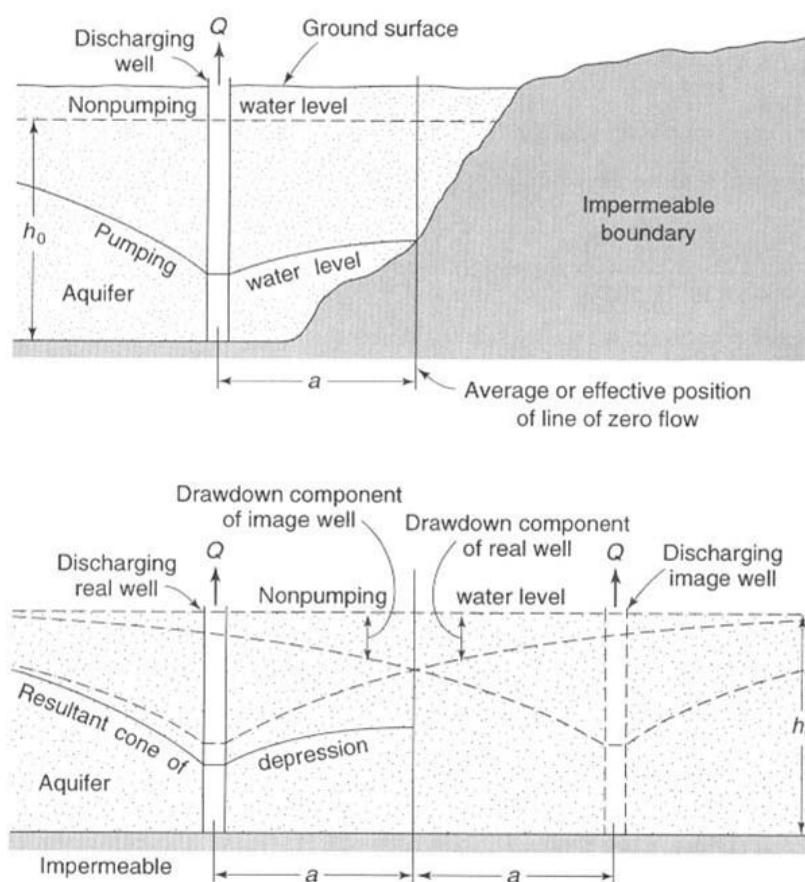


Figure 4.1. Sectional views of a discharging well near an impermeable boundary and the equivalent system using the method of images (Todd and Mays, 2005).

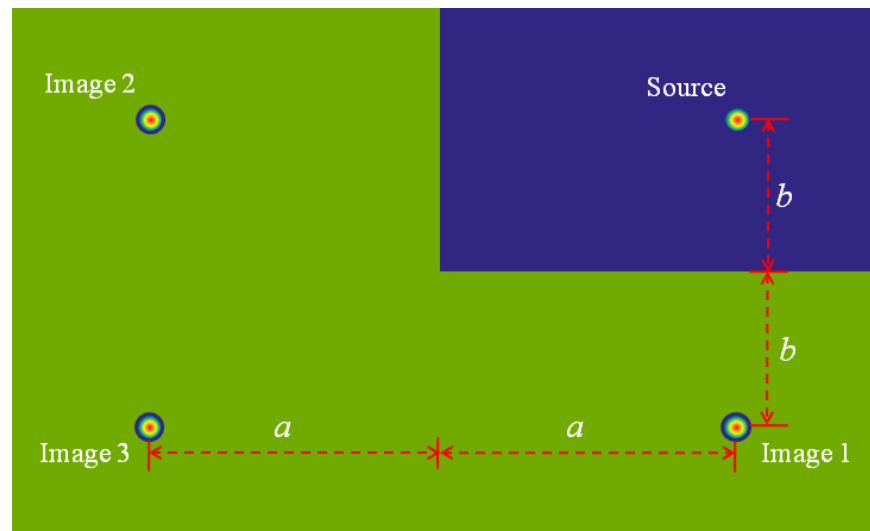


Figure 4.2. Typical 2-D layout of an image well system used in hydraulics of groundwater, where a is the perpendicular distance from the source to the vertical boundary and b is the perpendicular distance to the horizontal boundary.

In two dimensions, Image 1 and Image 2 in Figure 4.2 are the images with respect to the east-west boundary and the north-south boundary respectively; however, Image 3 is a representation of new reflections of Image 1 with respect to the north-south boundary and of Image 2 with respect to the east-west boundary. Without Image 3, the flow in the system would not be balanced and the gradient would not be zero at the boundaries.

The methodology can be explained to a greater extent by examining the example of a rectangular aquifer surrounded by impermeable boundaries (e.g. land) and recharge boundaries (e.g. a lake), in Figure 4.3.

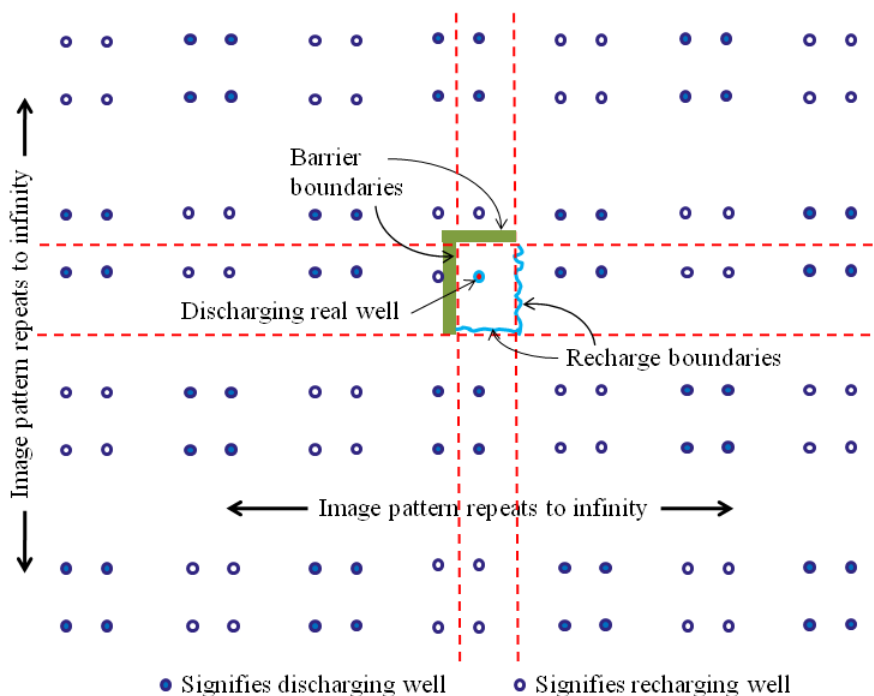


Figure 4.3. Plan of an image-well system for a rectangular aquifer (Adapted from Todd and Mays, 2005).

When two boundaries are parallel to each other, “analysis by the image-well theory requires the use of an image-well system extending to infinity. Each successively added secondary image well produces a residual effect at the opposite boundary.” (Todd and Mays, 2005). This effect will be progressively less as the system becomes balanced.

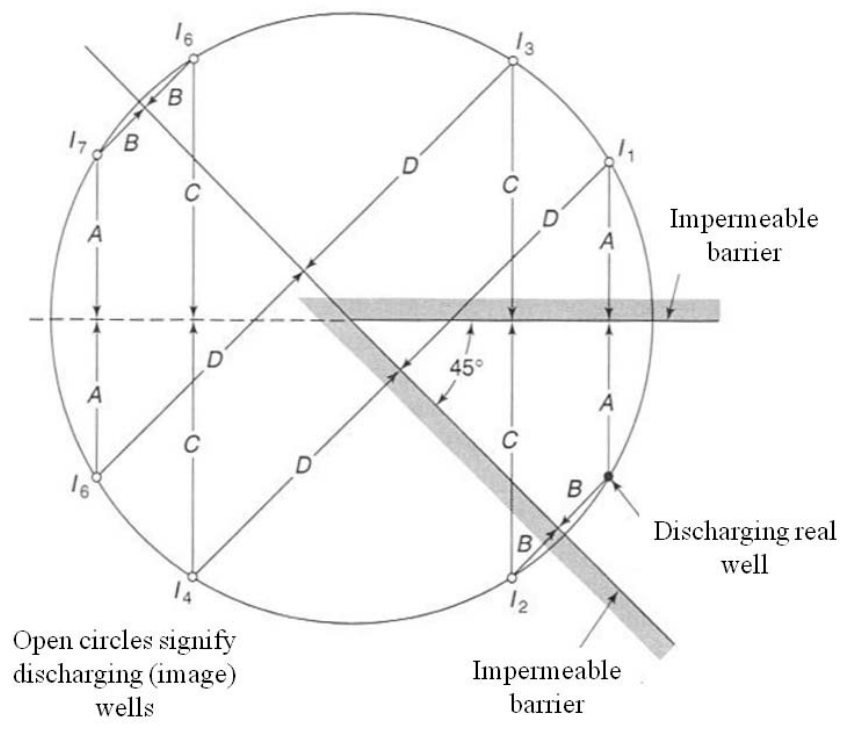


Figure 4.4. Image well system for a discharging well in an aquifer bounded by two impermeable barriers intersecting at 45° (Todd and Mays, 2005).

Figure 4.4 shows how the flow balance is attained when the boundaries are not perpendicular to each other. At 45° , the source and 7 image wells (I_i) must be considered for superposition.

4.2.2 Methods for the Adjustment of the Method of Images to Variable, Continuous or Discontinuous, Irregular, Approximately Curved Boundaries in Two Dimensions Applied to Gaussian Distributed Sources

Methods include the creation of single and general case scenarios, along with geometric approximations and engineered computed algorithms that matched such approximations. For the algorithms, basic trigonometric laws were employed. The Python programming language was used to automate all the procedures. The method of images presented in the previous section was taken in account at all times during the development of the method for finding concentration profiles subject to variable, continuous or discontinuous, irregular, and approximately curved boundaries in two dimensions applied to Gaussian distribution sources.

4.3 Discussion of the Method of Images to Provide Predictive Gaussian Distributed Sources with Variable, Continuous or Discontinuous, Irregular, Approximately Curved Boundary Conditions

The physical reality of the ocean boundaries (the shoreline) is approximated to an ideal condition in which a “relatively-flat ocean bottom” is truncated by a theoretically vertical barrier that is assumed to retain the sunken oil without compromising the oil’s mass conservation and the Fickian nature of the transport process; that is, it is assumed that phenomena that might occur close to a shoreline environment such as oil adsorption onto the barrier’s material or flow into the porous medium do not occur. Figure 4.5

replicates the physical approximation in one dimension. This is used as a numerical approach to calculate the effect of the vicinity of coastlines in oil spill scenarios.

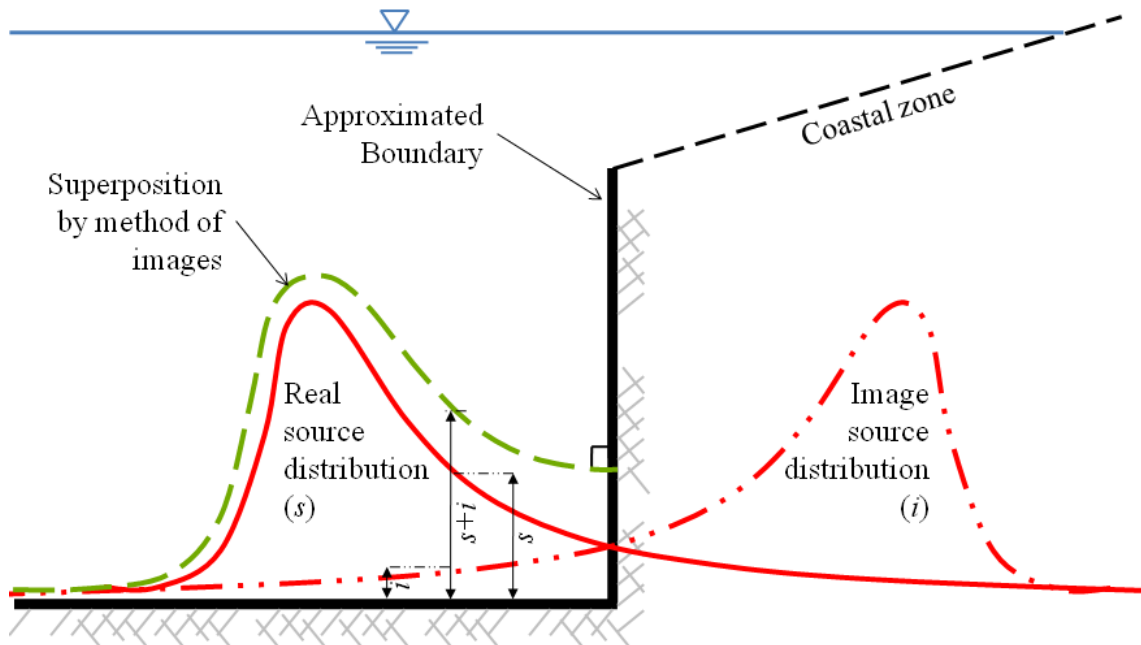


Figure 4.5. Graphical depiction of the Method of Images in one dimension applied to sunken oil distributions.

Consider an example in which an oil patch is denominated ‘the source’. The source occurs at a certain point $(x_0, y_0) = (x, 0)$ on the bottom of a bay and has a concentration distribution $c(x, y)$. This point lies at some distance x from the shoreline which, by definition, has a concentration gradient of zero. As the patch approaches the coastal zone, oil (a concentration hotspot or mass in a general case) at the leading edge begins landfall. It is then assumed that this mass is reflected back into the water, producing an

accumulation in the nearshore environment by the principle of superposition applied in the image theory. In one dimension the reflection is easily computed as a sum of the inflowing mass and the reflected mass. However, in two dimensions, when the shoreline is straight but does not form an orthogonal angle, reflections are not perpendicular with respect to one line or the other (or to only one line as in 1-D), and calculations become cumbersome.

Therefore, a new method for computing (relative) mass accumulation by setting up imaginary conditions to replicate the curve of the boundary is necessary. The method of images momentarily replaces the boundary by an “image source” or a “reflection”, equal in concentration distribution but located at opposite coordinates $(-x, 0)$ across the boundary (see Figure 4.5).

This new situation is equivalent to the original layout because superposition of the source and its reflection concentration distributions will simulate the zero concentration gradient at the boundary and will account for spreading and contention of oil. The result is then the summation of concentration distributions on the source side, and all results imagined to occur beyond the boundary can be ignored. Figure 4.5 depicts the concept.

In two dimensions, the method needs to consider that (1) the coastline has a curved shape and can be continuous or discontinuous, making the use of the bi-dimensional image systems shown in Figure 4.2 or Figure 4.4 by themselves, inappropriate; (2) trying to approximate the coastline to a system with multiple-consecutive orthogonal boundaries

or to a system with multiple-consecutive angles is possible, but highly demanding computationally, plus geometrically intricate because of the fact that each image must sequentially reflect with respect to all other existing orthogonal boundary (or contiguous angle in the case of consecutive angles) with the aim to attain mass conservation; (3) each modeling scenario in this dissertation is user-defined and the source(s) occurs within an unknown parameter space, therefore, using a general geometrical approximation of any set of boundaries is desirable; and (4) the Gaussian distribution that describes the unknown source(s) is geometrically advantageous because its parameters (known or within an algorithm that uses them as unknowns) can be combined and used along the reflecting method of images in a way that orthogonal or consecutive known angles are not necessary. The accommodation of the method of images to variable, continuous or discontinuous, irregular, approximately curved boundaries in two dimensions applied to Gaussian-distributed sources is explained next by the use of a single-segment boundary and then expanded to consecutive segments for better record.

4.3.1 Single line segment case:

Consider the reflection of the Gaussian *patch* shown in Figure 4.6. The boundary in this figure is sloped and the usual solution is to approximate it to two orthogonal boundaries and solve the case as in Figure 4.2. However, if the figure was rotated such that the line segment-shaped border becomes vertical or horizontal, the image will be automatically placed where it is shown in Figure 4.6, as it would be reflected in a mirror.

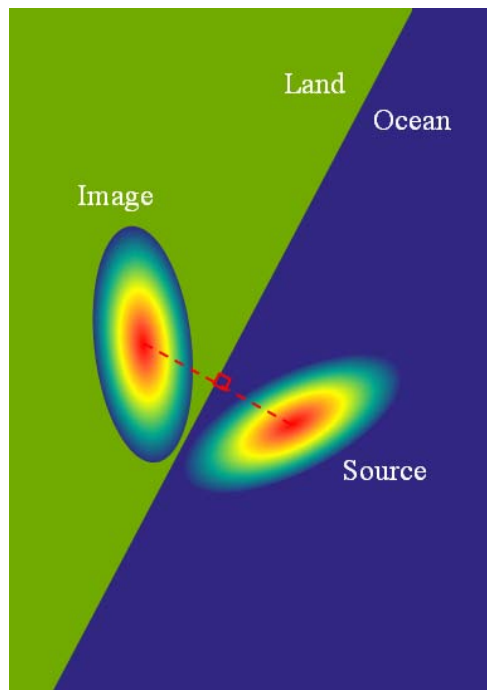


Figure 4.6. Reflection of a Gaussian *patch* across a sloped single line segment boundary.

Note that the shape of the source in Figure 4.6 is reflected including its direction (it is no longer a point source like it is assumed in well hydraulics). The reflection of the complete shape can be attained in two ways: (1) by reflecting every point (x_i, y_i) that composes the distribution with respect to the line segment, or (2) by using general trigonometry to (a) find the image of the Gaussian mean (μ_{x_r}, μ_{y_r}) , (b) find the correlation coefficient (ρ_r) that corresponds to the imaginary Gaussian patch, and (c) find the imaginary, distorted standard deviation in each direction $(\sigma_{x_r}, \sigma_{y_r})$ for the reflected patch. Both options are used in SOSim. The first is expected to occur automatically because every point in the unknown parameter space participates in the

Bayesian algorithm thanks to the baseline programming structure for use of combinatorial math. The second is implemented as a shape and mass conservation control technique and is described next.

The image of the Gaussian mean coordinate, (μ_{x_r}, μ_{y_r}) , is found using trigonometry, the source's mean (μ_x, μ_y) , and two points on the straight boundary section (x_1, y_1) and (x_2, y_2) , as shown in the Figure 4.7.

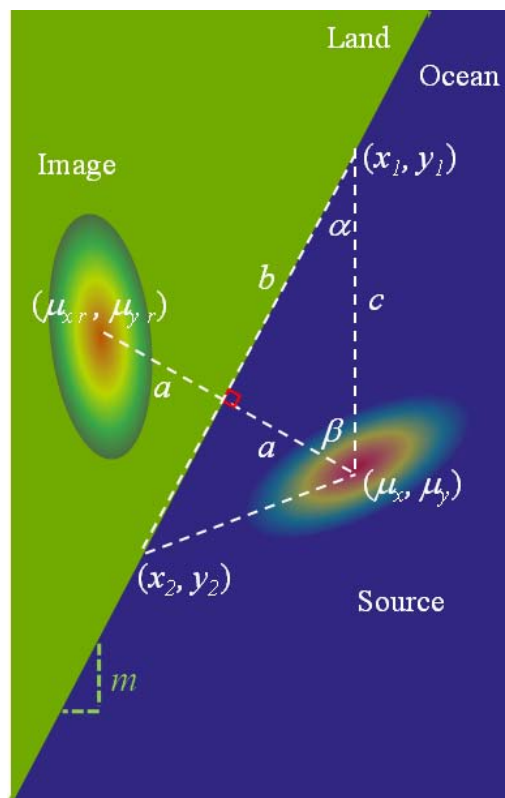


Figure 4.7. Geometry of the reflection of the Gaussian mean with respect to a single line boundary.

The solution in this case where the reflection takes place with respect with only one boundary is straightforward and evident from Figure 4.7. In the case of multiple and variable boundaries, where the concentration profile near boundary conditions depends on every boundary, the solution is demanding and requires a numerical algorithm, shown in Figure 4.8. In addition, consider the fact that the mean is an unknown, transient ($\mu = x_0 + vt$) parameter in the model of this dissertation, which makes the general solution dependent upon the baseline programming structure that uses combinatorial math among unknown parameters.

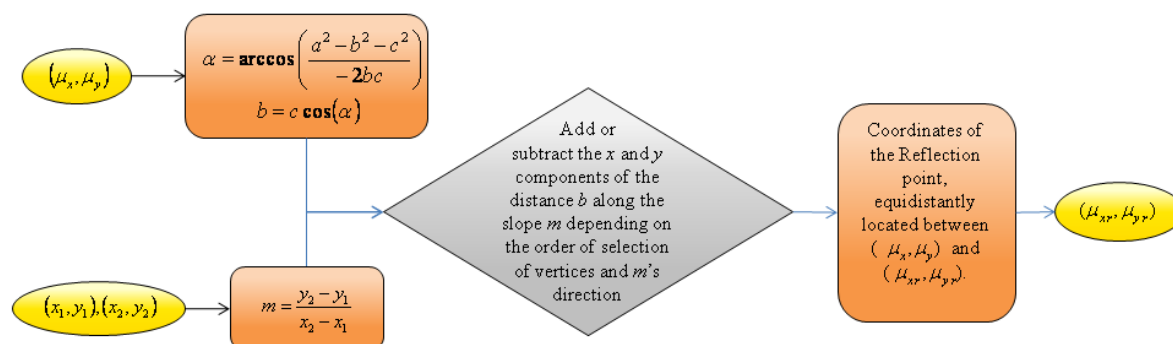


Figure 4.8. Algorithm to find the Gaussian mean of the imaginary Gaussian puff, (μ_{xr}, μ_{yr}) , from the mean of the source and two points of a line traced on the single boundary.

The standard deviation of the imaginary puff also changes with the reflection (although this may not be evident in many cases). This can be addressed by using the geometry employed to locate the imaginary Gaussian mean in Figure 4.7. Two points

analogous to the mean need to be reflected and then related to the standard deviations: $(\mu_x + \sigma_x, \mu_y)$ and $(\mu_x, \mu_y + \sigma_y)$. The standard deviations of the image are then $\sigma_{x_r} = (\mu_x + \sigma_x)_r - \mu_{x_r}$ and $\sigma_{y_r} = (\mu_y + \sigma_y)_r - \mu_{y_r}$. Figure 4.9 shows the graphical approximation.

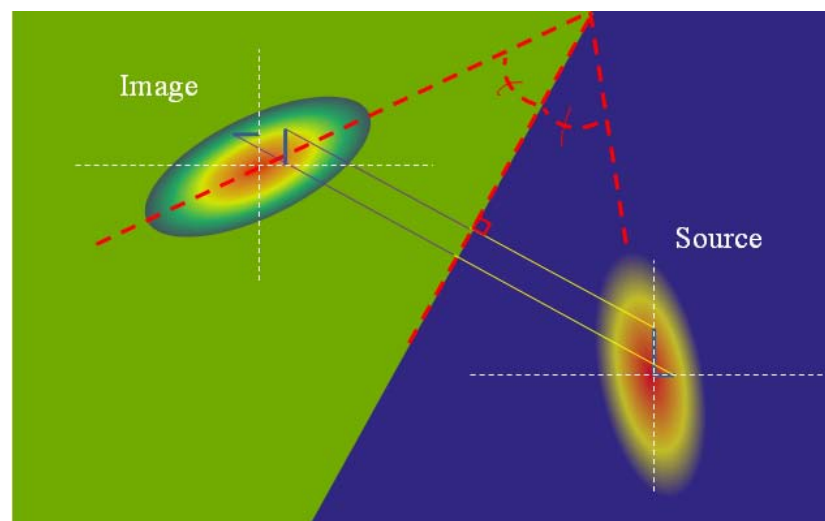


Figure 4.9. Approximation to the relationship between source and imaginary standard deviations.

The last parameter of a bivariate Gaussian concentration profile that is transformed by a reflection with respect to an inclined boundary is the coefficient of correlation. The coefficient of correlation is the statistical relationship of dependence between two (or more) random variables, in this case, between the X and Y directions of the Gaussian distributions in the Lagrangian system. The coefficient of correlation in a bivariate

Gaussian distribution is not a slope traced between the two directions, but it reflects the noisiness and the direction of the linear relationship (see Figure 4.10).

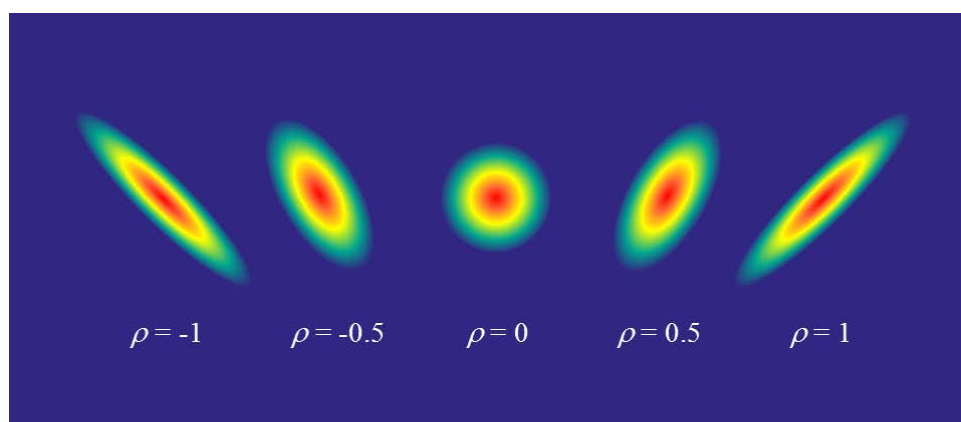


Figure 4.10. Set of bi-dimensional Gaussian profiles showing correlation coefficients between x and y directions. Note that the correlation coefficient reveals the direction and noisiness of the linear relationship (figure adapted from Wikipedia.org).

Resuming the reflection through a single line segment boundary case, the relative direction between the source Gaussian distribution and the land boundary (given by source correlation coefficient, ρ_s , and the slope of the boundary segment, m , respectively) is used towards finding the correlation coefficient of the Gaussian reflection, ρ_r , shown in Figure 4.11. The standard deviations of the imaginary Gaussian *patch* are confirmed in the Bayesian algorithm by use of the relationship $\sigma_{x_r}\sigma_{y_r} = Cov(x, y)/\rho_r$, where $Cov(x, y)$ is the covariance of the x and y directions given by $Cov(x, y) = E[(x - \mu_{x_r})(y - \mu_{y_r})]$, and ρ_r is the correlation coefficient of the image, found by means of the trigonometric algorithm shown in Figure 4.12.

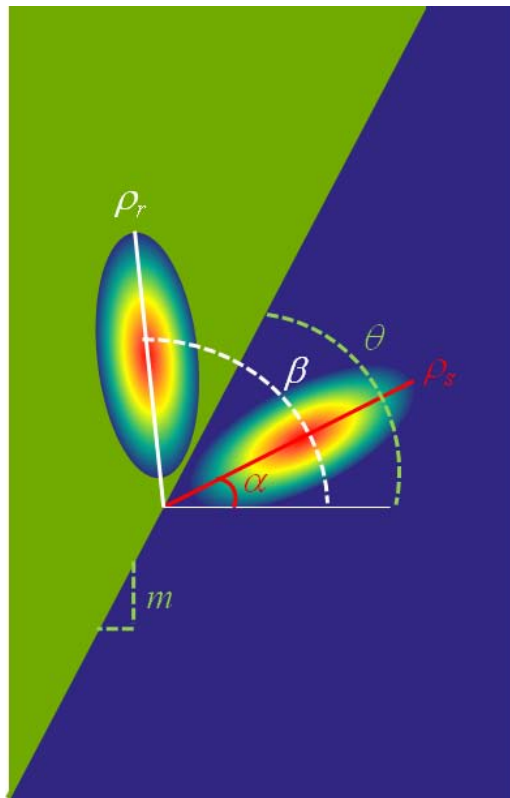


Figure 4.11. Relative directions of Gaussian source and single line segment boundary, used to find the parameters of the imaginary Gaussian puff.

The solution to the case shown in the general Figure 4.6 has been explained by the use of a single-segment boundary based on the advantages that the parameters and the geometry of a Gaussian distribution carry. Nevertheless, a general geometrical approximation of any set of boundaries is developed in the following section in order to automate and accommodate the method of images to variable, continuous or discontinuous, irregular, approximately curved boundaries in two dimensions applied to Gaussian-distributed sources.

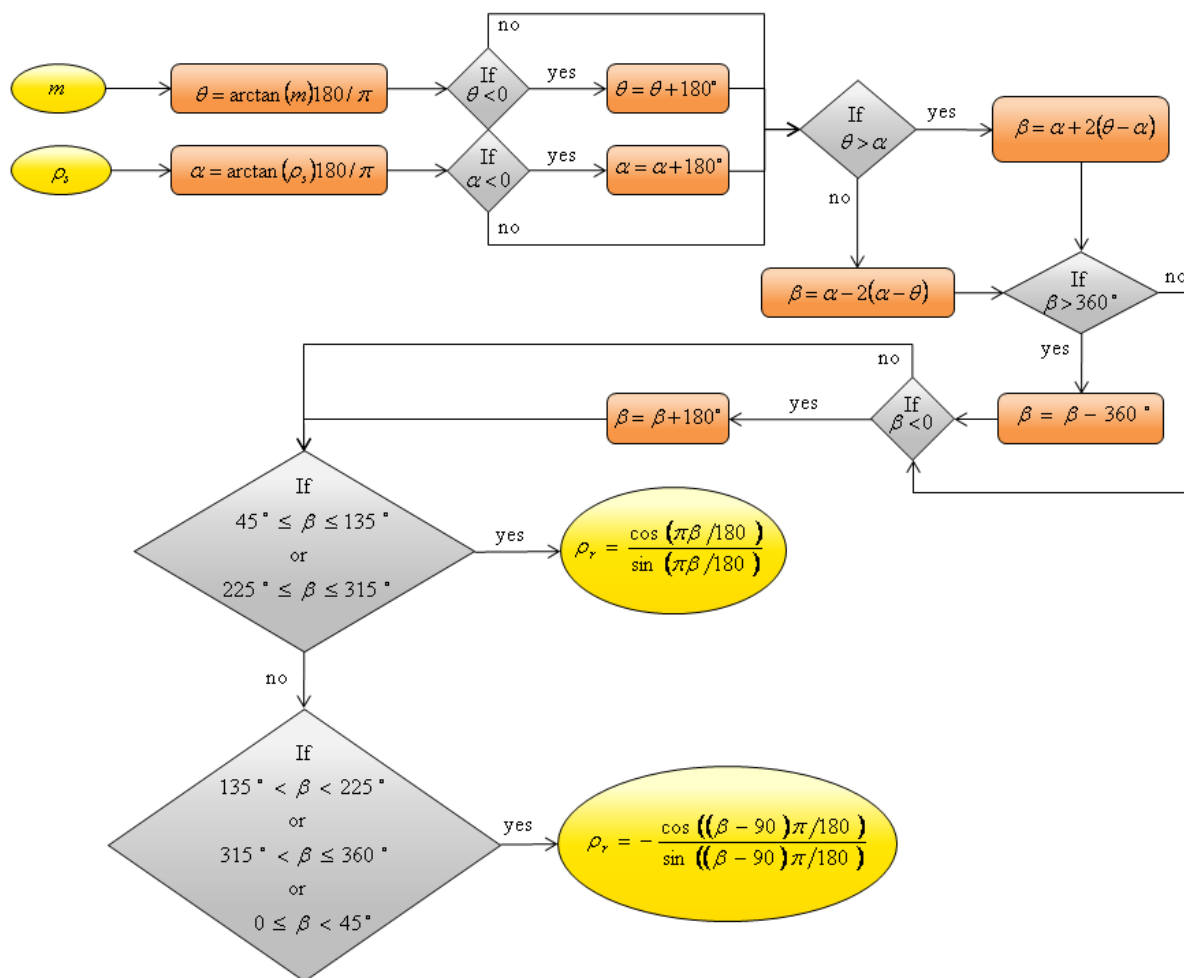


Figure 4.12. Algorithm to find the correlation coefficient of the imaginary Gaussian puff, ρ_r , from the slope of the single line segment boundary, m , and the correlation coefficient of the Gaussian source, ρ_s .

4.3.2 General case:

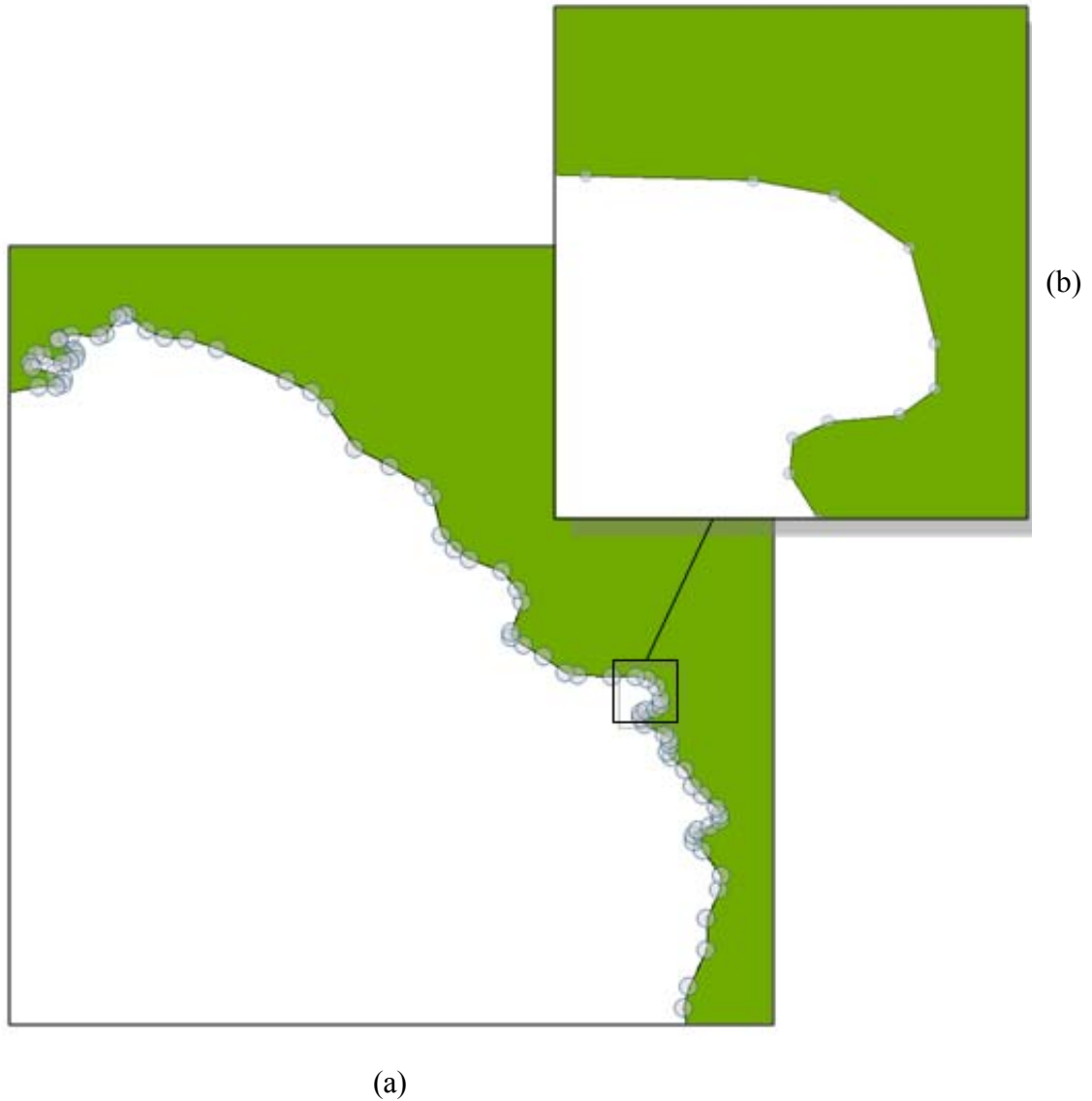


Figure 4.13. (a) A geographic land mass to the northwest of the Peninsula of Florida, on the Gulf of Mexico, is approximated by a set of continuous non-horizontal and non-vertical lines, joined by circular vertices. (b) Zoomed in bay that can be represented by a polyline of only 10 vertices.

Figure 4.13 shows how a curved boundary results from a continuous number of line segments with any slope (a polyline), that are superimposed to the shoreline. Similar to the internal boundary condition of a finite difference model, the shorter the lines, the better the precision of the method. The graphical dissimilarity here is that the curved boundary is approximated by a continuous polyline, not by a staircase-like set of vertical and horizontal steps. The adjustment of the image theory to curved boundaries is also believed (not tested in this dissertation) to be functional for discontinuous land masses given that reflection can work with respect to any and all lines. SOSim, however, was designed to incorporate only one continuous polyline, and does not have the functionality of calculating oil mass accumulation in areas such as shown in Figure 4.13.

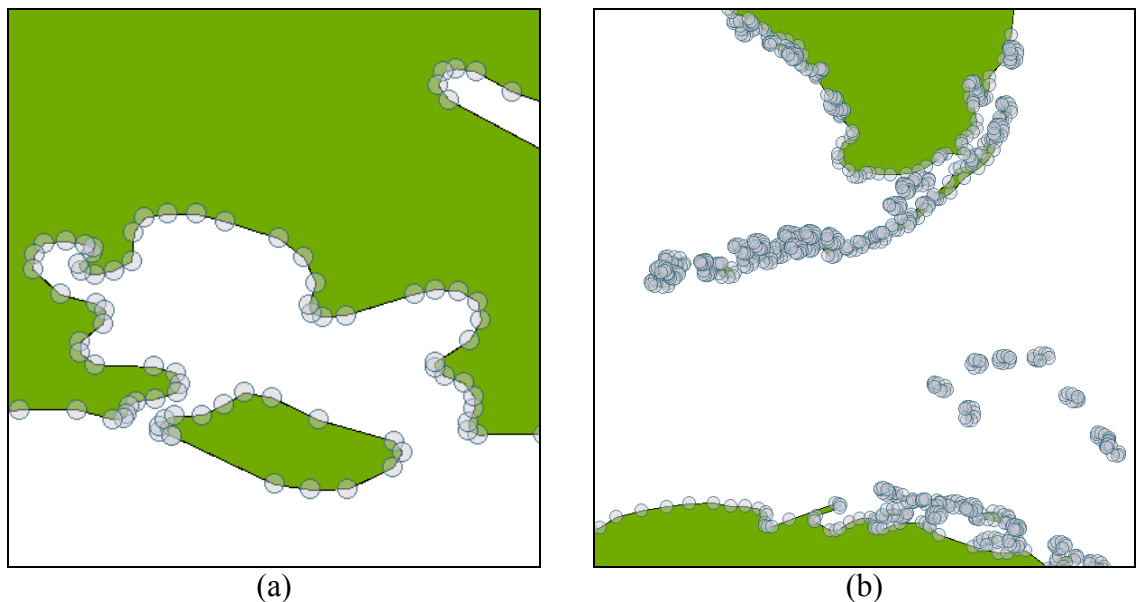


Figure 4.14. Geographic scenarios that can be approximated by polylines. Vertices are shown as circles. (a) Bay in Louisiana State, (b) the Caribbean waters between the southernmost point of Florida and the northern part of Cuba.

Figure 4.14 shows real example scenarios with discontinuous and/or multiple bi-dimensional boundary conditions which can be solved using the general procedure with more than one polyline.

In the single boundary case described above, no other boundary has effect on any of the sources, and thus there is a single reflection per source (or keeps reflecting to the same place in both sides); this case will conserve the mass, and is considered of zero order. When multiple boundaries coexist, reflections are not perpendicular with respect to only one line but instead it is assumed that one reflection can occur with respect to each perpendicular projection to a line. The boundary is interpreted as a series of connected lines, where every single source is reflected with respect to each of the line segments.

In the application of the method of images in groundwater hydraulics to modeling flow near aquifer boundaries, the source is reflected with respect to each horizontal or vertical aquifer boundary, and each subsequent reflection is in turn affected by the boundary that did not reflect it before (Bear, 1979, Todd and Mays, 2005), as shown in Figure 4.2, Figure 4.3 and Figure 4.4.

Another approximation to solve angled boundary conditions (Bear, 1979, Todd and Mays, 2005) is also used in groundwater hydraulics, which is restricted to a single angle at a time, for a given source (Figure 4.4). In every case, each of the image points are employed to feed one function that must be superimposed to the function of the source. The superposition process is no other than the summation of all the function as terms in

the final equation. The concentration profile of use is one which belongs to the modeling area (the aquifer in groundwater hydraulics, the sea in the case of this dissertation and SOSim).

The approach of considering perpendicular or parallel boundaries (that are not necessarily contiguous) is now adapted to non-orthogonal continuous or discontinuous line segments representing geographical boundaries to particle movement.

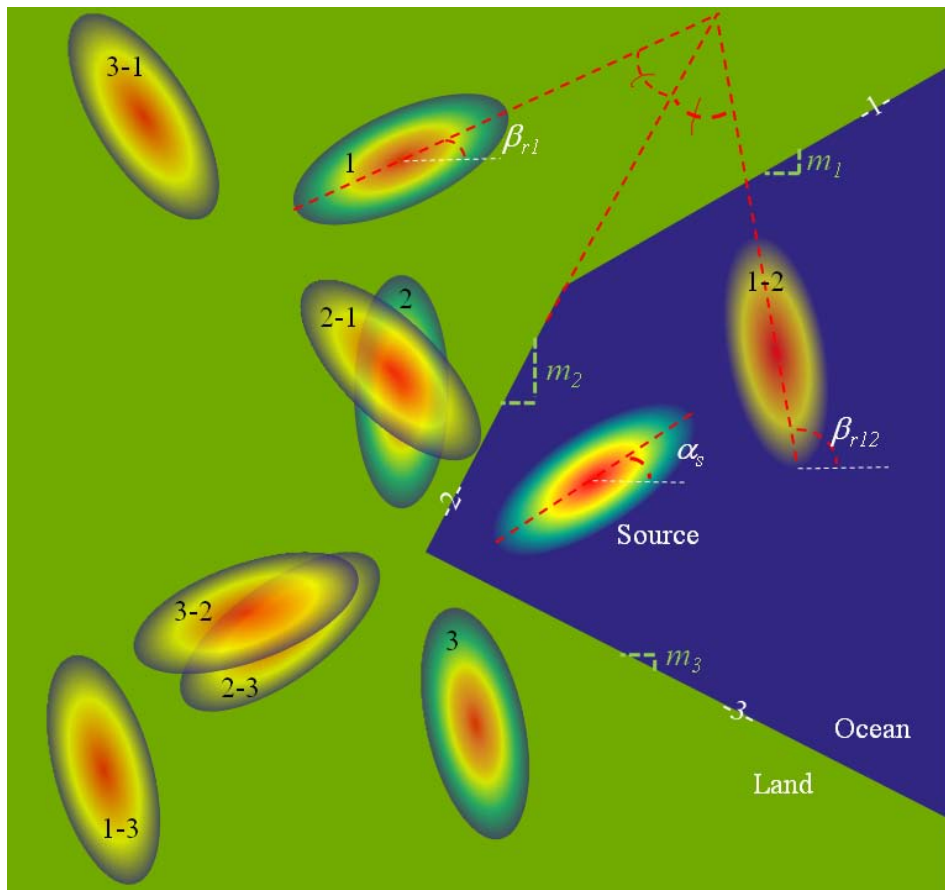


Figure 4.15. Hypothetical approximation to the reflection of a bivariate Gaussian distribution with respect to multiple, continuous line boundaries.

Figure 4.15 shows the approximate reflection of a bivariate Gaussian distribution with respect to multiple, continuous line boundaries. Note that lines are labeled -1-, -2-, and -3- from north to south. The source is first reflected with respect to -1-, -2-, and -3-, and the results of this first reflection are bluish patches labeled 1, 2 and 3. If the mass of the image is the same of its source, it is evident that a superposition will result in a mass imbalance between both land and ocean sides in this called “first-order reflection” as will be explained below. The imbalance is countered by (1) reflecting the bluish patch 1 with respect to all 3 lines, including with respect to the boundaries that did and did not reflect it before; (2) reflecting the bluish patch 2 with respect to all 3 lines, and (3) repeating the same for the bluish patch 3. The consequential images are 3 per bluish patch, are shown in yellowish color in Figure 4.15, and are labeled with double nomenclature, e.g. 1-2, which means a patch product of the reflection of the bluish distribution 1 with respect to the line boundary -2-. These yellowish images are called of “second-order reflection”.

Certainly, the geometry of the case shown in Figure 4.15 is unique. A model in which boundary conditions are to be selected by the user on a map of a graphical user interface needs a generalization of the geometry. Such generalization needs to be built based on only one required input: the coordinates of the polyline vertices and the mean of the concentration source (of transient and Bayesian unknown nature in the model developed in this dissertation). The variability of both the parameters of the Bayesian model and the boundary conditions for each scenario that needed to be modeled in SOSim gave way to the construction of a general geometry approximation with the objective of creating

computer algorithms for express calculation of the images' Gaussian parameters. The algorithms are described after presenting the repetitive nature of the reflection process and its relationship with the mass conservation principle.

Repetitive Nature of a Reflection Process and Mass Conservation

Let the correlation coefficient of the a Gaussian source or *patch* be ρ_s , and the correlation coefficient of its correspondent image or reflection be ρ_r (refer to Figure 4.11). Any time that the model considers a ρ_s of a source, an algorithm needs to calculate ρ_r for each of the applicable reflections to be superimposed. There will be as many ρ_r coefficients as line segments (K) are in the curve approximation (refer to bluish patches in Figure 4.15). In addition, as consequence of the superposition of the K Gaussian functions, the equation that counts the effects of the curved boundary conditions will have K terms for the first reflection order only (refer to yellowish patches in Figure 4.15).

During a reflection process, a source is reflected with respect to a boundary segment and, the produced imaginary reflection is mirrored back with respect to all available boundaries, in a way that the source and the imaginary sides are constantly experimenting mass imbalances that will occur to one side and then the other, until the total original mass of the source is equaled at both sides. Each time that a new reflection is needed in order to balance the mass, it is considered that a new *reflection order* is a need to add. Let

Figure 4.15 explain the concept of reflection orders. Based on the figure, in which the curved boundary condition is approximated to a total of 3 boundary segments, it can be observed that per each source s (or zero order term), there will be $K^1 = 3$ number of first order terms and $K^2 = 9$ number of second order terms. Generalizing, there will be K^n number of terms of order n per source function in a geographic condition approximated by $\Xi = K$ number of boundary segments. Notice that the increment of terms is not linear but follows a power law, which translates in the augmentation of computational demands as the reflection order increases. A bivariate Gaussian distribution represents each patch:

$$f_j(\mathbf{X}, t | \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, \rho_j) = \mathcal{N}(\mathbf{X}, t | \theta_j) = \frac{1}{2\pi\sigma_{x,j}\sigma_{y,j}\sqrt{1-\rho_j^2}} \exp\left[\frac{-Bm_j}{2(1-\rho_j^2)}\right], \quad (3.4)$$

$$\text{in which } Bm = \frac{(x_m - x_0 - \mu_x)^2}{\sigma_x^2} - \frac{2\rho(x_m - x_0 - \mu_x)(y_m - y_0 - \mu_y)}{\sigma_x\sigma_y} + \frac{(y_m - y_0 - \mu_y)^2}{\sigma_y^2}.$$

Equation (3.4) will adopt the resulting superposition until the mass is balanced by receiving all the patches as sum terms, as follows:

$$\mathcal{N}(\mathbf{X}, t | \theta_j) = \frac{1}{2\pi\sigma_{x,j}\sigma_{y,j}\sqrt{1-\rho_j^2}} \exp\left[\frac{-Bm_j}{2(1-\rho_j^2)}\right] + \sum_{n,k}^{N,K} \frac{1}{2\pi\sigma_{x,j,n,k}\sigma_{y,j,n,k}\sqrt{1-\rho_{j,n,k}^2}} \exp\left[\frac{-Bm_{j,n,k}}{2(1-\rho_{j,n,k}^2)}\right], \quad (4.1)$$

in which

$$Bm_{n,k} = \frac{(x_m - x_0 - \mu_{x\ n,k})^2}{\sigma_{x\ n,k}^2} - \frac{2\rho_{n,k}(x_m - x_0 - \mu_{x\ n,k})(y_m - y_0 - \mu_{y\ n,k})}{\sigma_{x\ n,k}\sigma_{y\ n,k}} + \frac{(y_m - y_0 - \mu_{y\ n,k})^2}{\sigma_{y\ n,k}^2}.$$

The total number of reflection orders N is achieved when there are negligible mass imbalances between the land and the ocean modeling areas. The repetitive nature of the reflection process towards mass conservation is solved by the use of iterative algorithms as described before in this dissertation and as explained in detail in the section below.

Modifying the Gaussian Parameters

For the theory of images to work for multiple, continuous or discontinuous line boundaries as shown in Figure 4.15, the parameters of the Gaussian source(s) need to be transformed multiple times in an iterative manner by using general numerical algorithms that find their base in the previously explained case of a single linear boundary. The first parameters to transform are the Gaussian means, (μ_x, μ_y) . Figure 4.7 is comparable to Figure 4.16, in which the geometry of the first-order reflection of the Gaussian mean is shown for the analogous case of a 4-vertices polyline boundary.

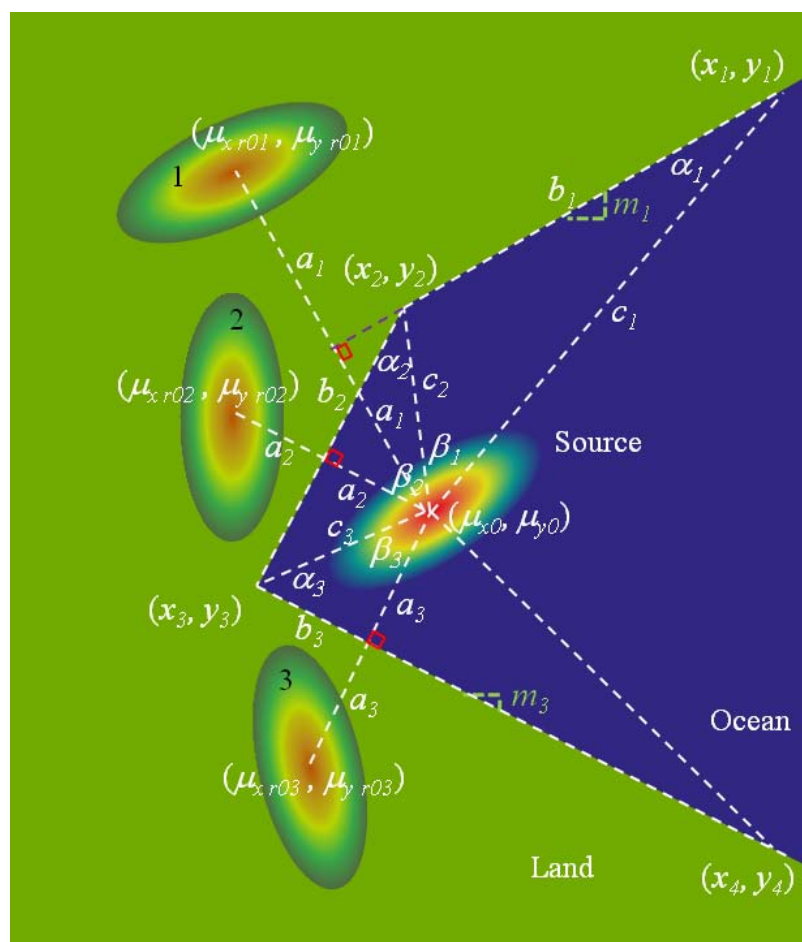


Figure 4.16. Geometry of the first-order reflection of the Gaussian mean with respect to multiple line boundaries.

Statistical means of the bluish imaginary *patches* must be found before a second-order reflection can be performed. One cycle per each first-order image 1, 2 and 3 must be performed following. Figure 4.17 tries to make clear how the second-order cycle works for the first-order image 3.

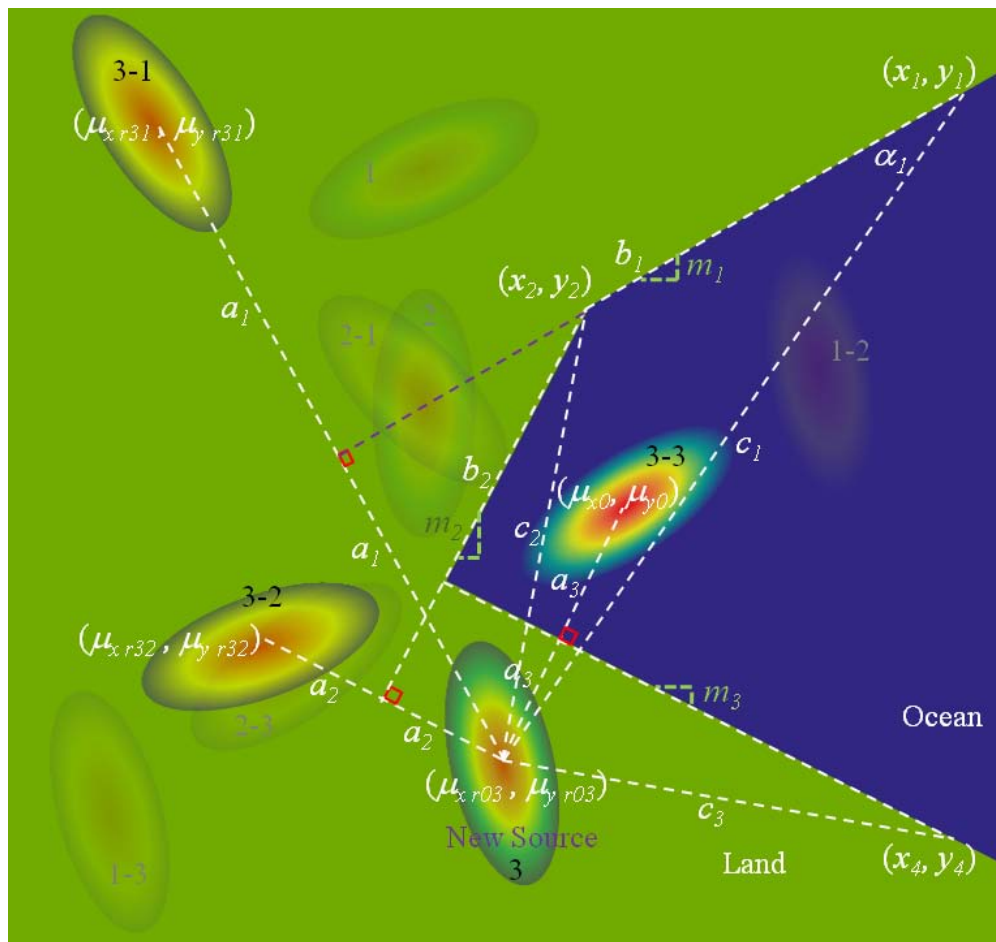


Figure 4.17. Example geometry of one of the second-order reflection cycles to obtain imaginary Gaussian means with respect to multiple line boundaries.

Although Figure 4.17 shows only one of the second-order cycles, it can be concluded from the plot that the process is cumbersome and requires a general geometry approach that has been developed in this dissertation. Figure 4.18 summarizes the iterative algorithm programmed to control the production of the Gaussian means of the successive line-based and ordered reflections.

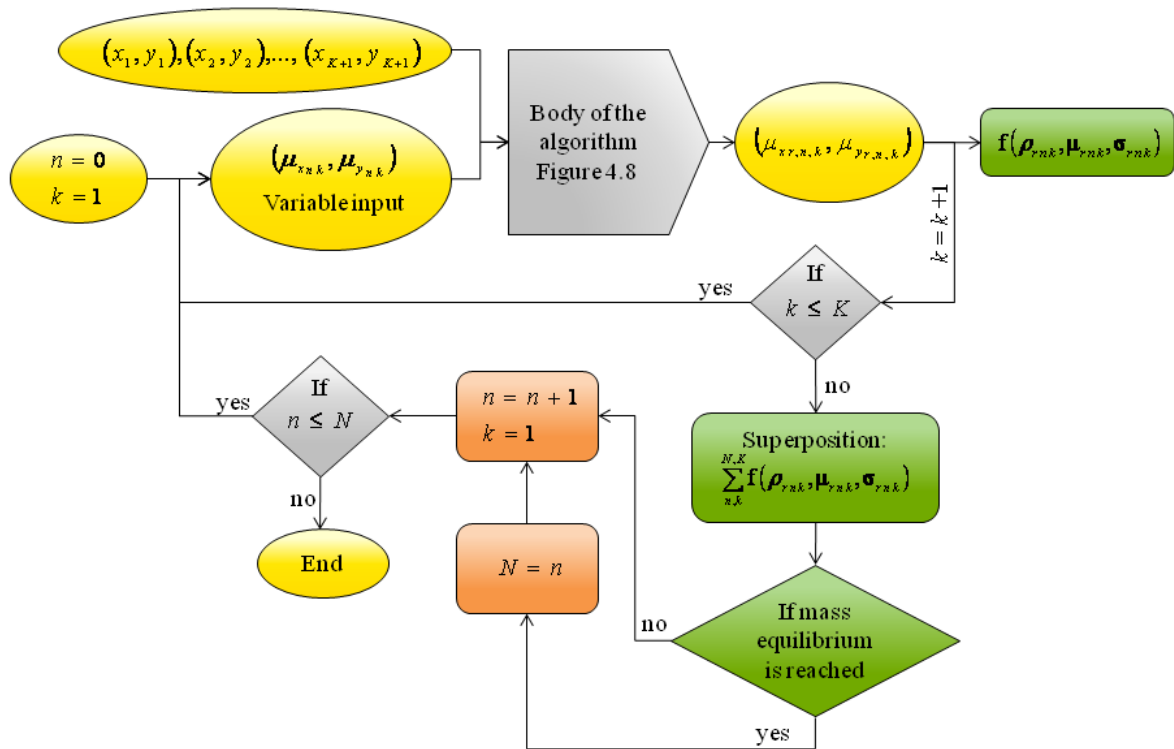


Figure 4.18. Iterative algorithm to control the production of the Gaussian means of the successive line-based and ordered reflections.

As in the single-line boundary case, the standard deviations of the imaginary puffs are also different, although not noticeably sometimes. This is addressed by using the same geometry of Figure 4.7 for each reflection and the iterative algorithm employed to locate the imaginary Gaussian means (Figure 4.18) but with the pairs $(\mu_{x\ n,k} + \sigma_{x\ n,k}, \mu_{y\ n,k})$ and $(\mu_{x\ n,k}, \mu_{y\ n,k} + \sigma_{y\ n,k})$ as the variable input, instead of $(\mu_{x\ n,k}, \mu_{y\ n,k})$. The output is analogous to the standard deviation vector by $\sigma_{x\ r,n,k} = (\mu_{x\ n,k} + \sigma_{x\ n,k})_r - \mu_{x\ r,n,k}$ and $\sigma_{y\ r,n,k} = (\mu_{y\ n,k} + \sigma_{y\ n,k})_r - \mu_{y\ r,n,k}$. His result is controlled in the Bayesian calculations

by testing the covariance of the *patches*. It is important to have in mind that in the predictive Bayesian model developed in this dissertation all parameters are unknown, and therefore all the algorithms are operated under the light of a baseline programming structure that allows for combinatorial math among the complete domain of the parameters.

The modification of the Gaussian parameters due to the application of the method of images to approximately curved boundaries is not complete without addressing the adjustment of coefficients of correlation of all possible reflections. Refer to the upper-right part of Figure 4.15, where the source has an angle α_s with the horizontal, the first-order reflection 1 has a resultant angle β_{r1} with the horizontal, and the second-order reflection 1-2 has a resultant angle β_{r12} with the horizontal. Analogously with Figure 4.11, α_s is the angular direction of ρ_s and β_{r1} is the angular direction of ρ_{r1} , where the last is found by using the algorithm in Figure 4.12 with ρ_s and m_1 as input. Similarly, β_{r12} is the angular direction of ρ_{r12} , which is calculated by using the algorithm in Figure 4.12 with ρ_{r1} and m_2 as input. By inspecting the procedure described it can be observed that the algorithm must be used in a repetitive, controlled way across each boundary line and each order of reflection in an iterative fashion explained with Figure 4.19.

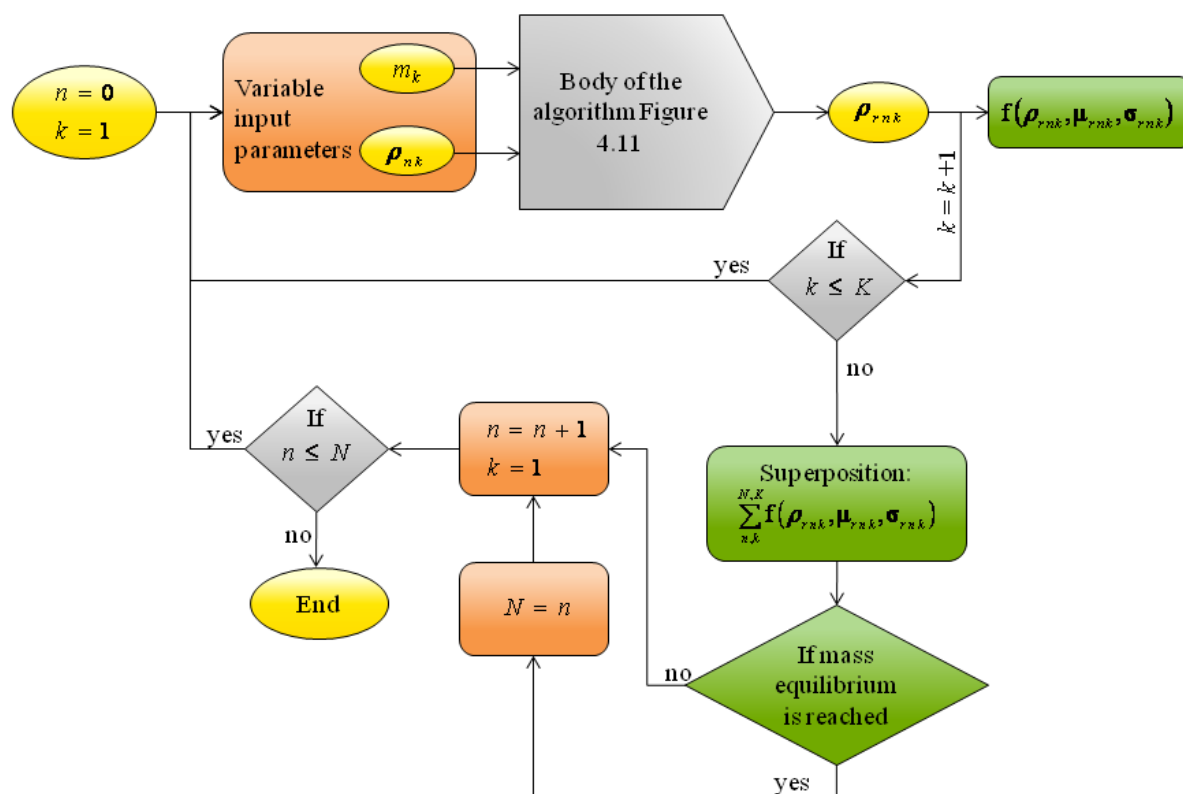


Figure 4.19. Iterative algorithm to control the production of the correlation coefficients of the successive line-based and ordered reflections.

The computer program contains all the algorithms described previously, embedded in the Core module. The activities in green shapes included in Figure 4.18 and Figure 4.19 are main processes within the Bayesian method with combinatorial structure. In SOSim, however, the iterative process is truncated after counting the second reflection order towards superposition (Equation (4.1)) because of computational run time limitations; nevertheless, tests demonstrate that the shape of the prediction and the mass balance are adequate and the method can be used in other applications.

Start and stop of the reflection process

There are two different start and stop rules to consider in the process. The first start-stop rule is related to the times at which the source function starts and finishes a superposition with an imaginary function; and the second rule is to manage the appropriate time to stop the iteration process, which is the time when mass balance has been reached.

The algorithms of Figure 4.18 and Figure 4.19 have a starting time, t_0 , that is accomplished only when the mean of a source is located at a distance of two standard deviations ($\pm 2\sigma$) from any boundary segment, close enough to produce an imminent image across the boundary and commence a superposition process. Because the Gaussian tail is mathematically infinite, the superposition process starts as the first tails of source (traveling inland) and imaginary functions (traveling outland) begin to overlap. The time to stop the reflection process, t_f , obeys to the same conditional but now corresponds to the time in which the tails of the functions no longer overlap (subject to the magnitude of $\pm 2\sigma$), after which the gradient zero at the boundary will persist without the need of superposition. In the Sunken Oil Simulator (SOSim), both the mean and the standard deviation are vector unknown parameters, reason why this start-stop rule was immersed in the Bayesian algorithms. When the prediction time is passed t_f , the algorithm is automatically aborted by means of an *a priori* estimation of the time of landfall.

The time at which consecutive reflection orders no longer create imbalances in the total mass between the modeling area and the imaginary inland portion is designated as the ideal time to stop the iterative process. The methodology developed includes an automatic decision algorithm that checks for imbalances and decides whether or not to proceed with calculations of subsequent reflection orders. In the software SOSim, nevertheless, it was found that the sum of second and lower order reflections provided sufficient detail to obtain an observable mass balance in the water body, and third and higher order reflections can be neglected, greatly reducing computational demands. Figure 4.20 shows a test scenario in which a single Gaussian source hits the coastline, drawn as a four-vertex (red points) polyline (green line) that approximates the geographical region shown in Figure 4.20 (a). The land portion has not been colored green as in other figures with the aim of allowing for the review of the imaginary portion of the result. Figure 4.20 (b) shows a Gaussian patch incoming to the coastline when no boundary conditions are in effect (no accumulation in time or mass conservation), figure (c) shows the result of the new methodology after superposing all the zero and first-order reflections with respect to the complete boundary, and figure (d) shows the profile after accounting for the complete zero, first and second-order reflection processes.

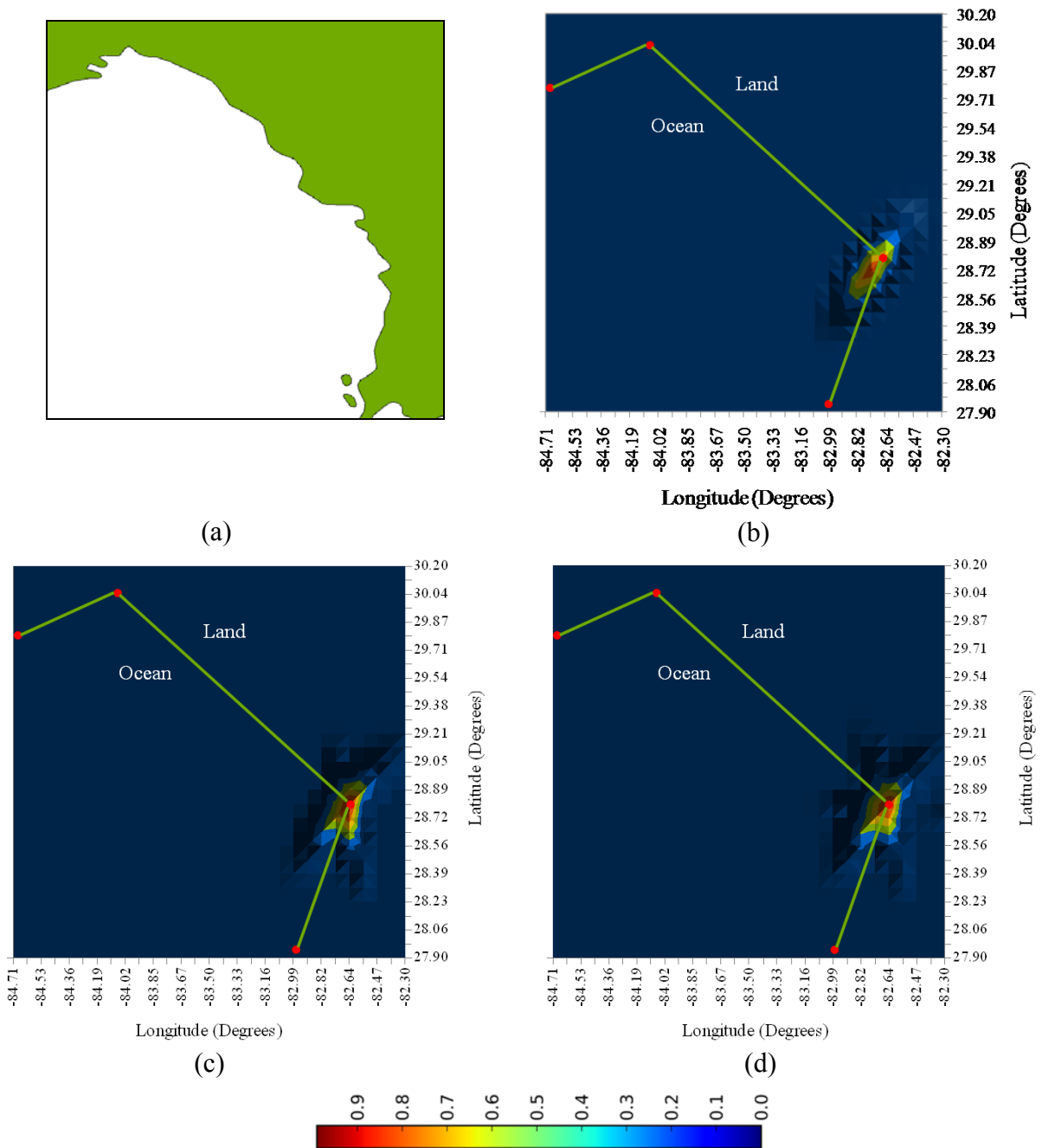


Figure 4.20. Relative mass distribution among ocean and the imaginary land counterpart for a Gaussian source that hits the coastline shown in (a), approximated by polylines when (b) no boundary conditions are in effect, (c) after accounting for zero and first-order reflection processes, and (d) after accounting for zero, first and second-order reflection processes.

Note that the differences between Figure 4.20 (c) and (d) are minimal at the shown prediction time, which suggests that a reasonable mass conservation has been attained only after the zero and first order reflection process. Accumulation tests can also be performed because the method developed is a dynamic system in nature, given the transient character of the Gaussian mean and the standard deviation worked with in this dissertation.

4.4 Conclusions

The analysis presented indicates that the new variation of the method of images developed to allow assessment of pollutant locations using predictive Bayesian Gaussian models given variable, continuous or discontinuous, irregular, approximately curved boundary conditions has been successful. The new method allows the effect of boundary conditions to be modeled even under the restrictions implied by the lack of known parameters. In addition, the use of computer programs eases the exercise of intricate geometries generalized in numerical algorithms, opening the possibility of modeling complicated curved boundary conditions as desired.

Chapter 5. SOSim System Development, including a Combinatorial Algorithm for the Computation of Multi-Modal, Highly Dimensional Stochastic Functions

This chapter describes the SOSim development from the point of view of the software design, including the programming environment chosen, algorithms established for implementation of the Bayesian approach and the model functionalities for the creation of the operative, processing, and graphic interfaces.

5.1 Python: The Programming Language

The Python programming language was chosen for development of the SOSim model based on the following criteria:

- The programming software is non-proprietary or non-commercial and open source,
- The programming environment supports relatively rapid prototyping and testing,
- The language is robust in terms of functioning in various operating environments with minimal damage, alteration or loss of functionality, and
- The source code is viewable at all times.

That is, by choosing Python, the resulting versatility of the model can lead to a generalized use any by public and by agencies that perform emergency response to spills in an expedited way.

Python is a general-purpose, high-level programming language first released by Guido Van Rossum in 1991 in The Netherlands. Python is flexible, applicable to many programming domains and available on many platforms. The language has an open, community-based development model managed by the non-profit Python Software Foundation (PSF). Python's design philosophy emphasizes programmer productivity and code readability (core syntax and semantics are minimalistic, employing white spaces and easy commands). Python supports multiple programming paradigms (primarily object oriented) and features a fully dynamic type system and automatic memory management, similar to Perl, Ruby, Scheme, and Tcl.

The Python programming language has an open, general public license (GPL) that allows source code modification, addition of original code as “imported” packages, and publication of derived work (as GPL). Packages that are created around the world are available to the Python community. Python has also been used as an extension language for many existing systems, including GIS, Web programming, security systems, data bases, numerical tools, and the R statistical package.

The SOSim model, although using and relying for much of its functionality on several existing Python packages and modules, consists of three principal Python modules developed as part of this dissertation: the graphical user interface (GUI) module, the operating and processing interface (OPI) module, and the core module. The GUI module (ui_SOSim.py in the SOSim source code) automatically lays out and retains characteristics of widgets, labels, canvases and templates in the graphical user interface,

retains raw user input, and imports Windows palettes and display. The OPI module (SOSimOPI.py in the SOSim source code) is the executable file. It imports and links all other modules, captures input information entered by the user in the GUI module, and operates interrelated buttons and activities of the GUI; it filters, organizes, and processes the input; passes ready-to-use variables and attributes to the core code; accepts modeling results back from the core module; processes the results; and sends display signals to the canvas layout of the GUI module to allow it to depict relative sunken oil concentrations on a map for further user interaction. It also controls modal behavior of the main windows, pop outs, menus and toolbars. The core module (SOSimCore.py in the SOSim source code) uses variables and attributes passed by the OPI module to compute the predictive Bayesian relative concentrations, saves output files, and passes results of the Bayesian process back to the OPI module for display by the GUI or for further use. The GUI has the capability to communicate with the core module through the OPI module during a model run, after a predictive result has been presented, to allow the user to request the modeling of contiguous or other areas as needed.

5.2 Methodology for Software Development

The SOSim software is in essence a PyQGIS - PyQt application. PyQGIS are the Python bindings of the Quantum Geographic Information System (QGIS) software, consisting of three modules written in Python (QGIS Organization, 2009). QGIS is an official project of the Open Source Geospatial Foundation (OSGeo), distributed under the GNU General Public License. The main components of QGIS are the GDAL library and

the GRASS geographic information system, both written in Python. QGIS is multi-platform and supports numerous vector, raster, and database formats and functionalities. The use of the PyQGIS bindings allowed the researcher to use all the functionalities of QGIS through scripting in Python, to build an original interactive GIS computer application, one of the contributions of this dissertation. The major result of using PyQGIS in SOSim is its interactive map canvas that is embedded in PyQt as a widget.

PyQt are the Python bindings of the Qt project, a C++ toolkit for cross-platform application development owned by Nokia. Like some versions of the Nokia's Qt, PyQt is open source, free software developed by the British firm Riverbank Computing and distributed under a variety of licenses including GNU General Public License (GPL) and commercial license. PyQt is implemented as a Python extension module, in such a way that it can be incorporated into a customized Python installation and used from the Python console or any editor. Computer applications developed in PyQt have the capability to support Linux, Unix, Mac OS X and Microsoft Windows. Version 1.0.2 of PyQGIS under OSGeo4W Kore distribution and version 4.4.3 of PyQt were used to develop the SOSim model. GUI programming aids were adapted from Summerfield (2007) and online developer communities (QGIS Organization, 2009; Qt Development Frameworks, 2009).

SOSim is composed of three new Python modules: the graphical user interface (GUI) module, the operating and processing interface (OPI) module, and the core module. In the GUI module, the graphic method offered by the Qt Designer was employed. Qt Designer

is the Qt's graphic utility to sketch main windows and dialogs and to layout the widgets in it (Qt Development Frameworks, 2009). The OPI module used in its algorithms several Python classes and methods of a collection of existing modules, including the called Sys, Os, Re, Math, Numpy, Calendar, String, Time, Shutil, Pickle, PyQt4, and QGIS. The SOSimCore module used methods from classes in the modules Math, Future, Numpy, Time, Itertools, Pickle, and Matplotlib with the qt4agg backend. The executable binding used the Py2exe and the Sip modules. The application was packed into a binary installer using the Inno Setup module.

The SOSim computer application falls in the category of software derived from Python, PyQt and PyQGIS among others. As such, the Python Software Foundation requires that the application developed be open source and have a GPL license (PSF, 2001). The terms for distribution, expansion and derivation of new tools from the current version 1.0rc1 of SOSim are included in the executable and source code distribution as required by law (Free Software Foundation, 2007).

5.3 Algorithm Development in SOSim Modules

SOSim is developed in three new Python modules: the graphical user interface (GUI) module (SOSimGUI.py), the operating and processing interface (OPI) module (SOSimOPI.py), and the core module (SOSimCore.py). The three modules interact and are connected as shown in Figure 5.1. The objectives, algorithms, and output features of each of them are described in this section.

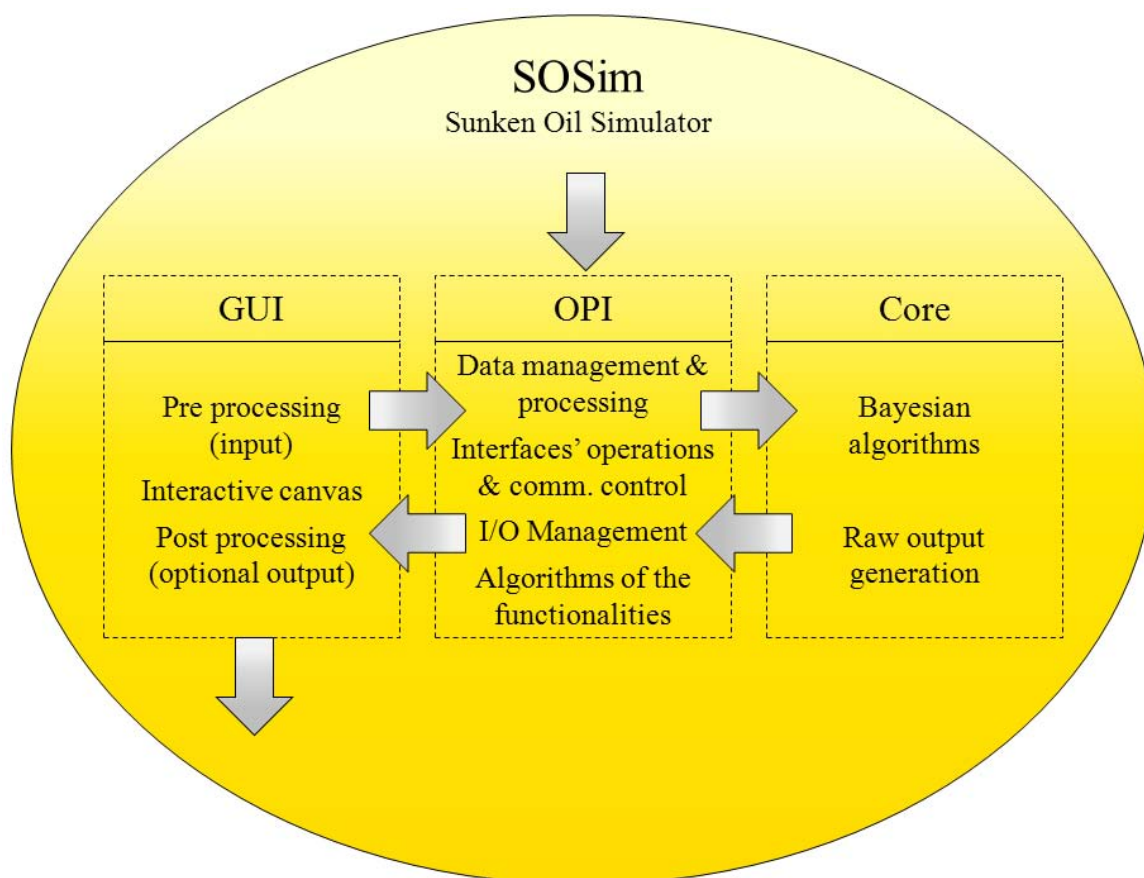


Figure 5.1. SOSim composition and interaction between modules.

5.3.1 SOSim GUI: Graphical User Interface

The graphical user interface (GUI) maps the output predictive relative concentrations geographically, presenting to the user the inferred location of sunken oil as a function of time, while controlling, with user input, the modeling area and shoreline boundaries used in the computations.

A preview of the GUI is shown in the Users Manual in Appendix A. The GUI was developed using the Qt Designer (as described in the methods section) and code embedded in the initialization *method* of the OPI module. The GUI contains three basic layouts other than the main toolbars and menus: (1) the pre-run layout or input section, where the user is prompted for input information concerning the spill, sampling campaign(s) and time(s), land boundary, desired modeling area, and prediction times, (2) the canvas layout, where results are displayed in interactive, georeferenced maps, and (3) the post-run layout or output section, which contains tools that allow the user to display the results at different times of prediction, run the model for contiguous areas, save results, print images, and perform other formatting tasks.

The GUI is the point of connection between the user and the OPI module; this last performing all required processing of raw input from users and providing the core module with the appropriate information. Among the most important tasks of the GUI, controlled through the OPI module, are:

- Collection of basic information on the spill in terms of the time of occurrence and coordinates (longitude and latitude) of the most significant loss;
- Display of the location of the spill on the map canvas within a 2-degree radius around the spill to allow selection of the desired modeling area;

- Assignment of the resolution and scale (rectangular) of the results based on default values or a user-defined number of modeling nodes in each direction: north-south and west-east, within the desired or default modeling area;
- Setting of customized dates and times at which sunken oil mass prediction is desired,
- Acceptance of output from the core module and geo-referenced by the OPI module, for display of it on the local map; presentation of relative probabilities of finding sunken oil, (0-1), on a relative, color-coded scale, with hotspots in red (1) fading to blue (0). Although probabilities of sunken oil decrease in time, the hue is re-rendered in every time calculation, such that the areas with the highest probability of finding sunken oil are shown in red at each modeled time, independently of relative probabilities shown for other times of prediction; and
- Display of post-run results at different prediction times and for contiguous modeling areas.

Detailed information on the operation of each of the features just described can be found in the Users Manual (Appendix A).

5.3.2 SOSim OPI: Operating and Processing Interface Module: Internal Processing of Input

The OPI module of SOSim is the largest module in terms of length of code and productivity, but it is not the most demanding of processing power. The OPI module is also the messenger and translator between the GUI and the Core modules. Detailed mathematical algorithms were created and implemented in the OPI module of SOSim to convert user input into programmable structures, process input as needed, generate local and global variables for all module's processes, compile graphical interactions, operate interrelated buttons, actions, signals and toolbars of the GUI, and process/control information needed for the predictive Bayesian calculations. These algorithms are several, and are included as *methods* in two Python *classes* developed in SOSimOPI: *SOSimMainWindow* and *DrawPolylineMapTool*. Each *class*, by definition of object-oriented programming, contains methods. The term *methods* refers to functions that perform unique activities to define and to provide different actions to its class, to be used by an extra parameter which is the object that it is to run on, and which belongs within the same class. The algorithms are extensive and vary in complexity. The most important are listed below.

- Interrelated and single operation of signals, actions, dialog boxes, menus and toolbars;

- Maintenance of scale and units across statistical calculations; plotting and interactive tools and devices;
- Operation of the interactive tool to assign boundary conditions, translation into mathematical expressions, and solving for geometry of boundary conditions;
- Differentiation of types of oil and calculation of the slick's sinking retardation;
- Storage and processing of input data from single or multiple sampling campaigns;
- Processing, storage, and passage to the core module of user input;
- Processing of results from the core module into maps and processing of optional post-run operations, which include management of prediction times and performance of partial recalculations to view results for contiguous modeling areas. Useful documentation functions including saving the results as images and printing are also included in the software under the OPI module.
- Uploading, saving, and using different files and images on disk (I/O) including calibration files; management of variable and default values, structures and map registry, relative time and locations management; and post-run functionality.

Unit Conversions: Geographical Versus Modeling Units

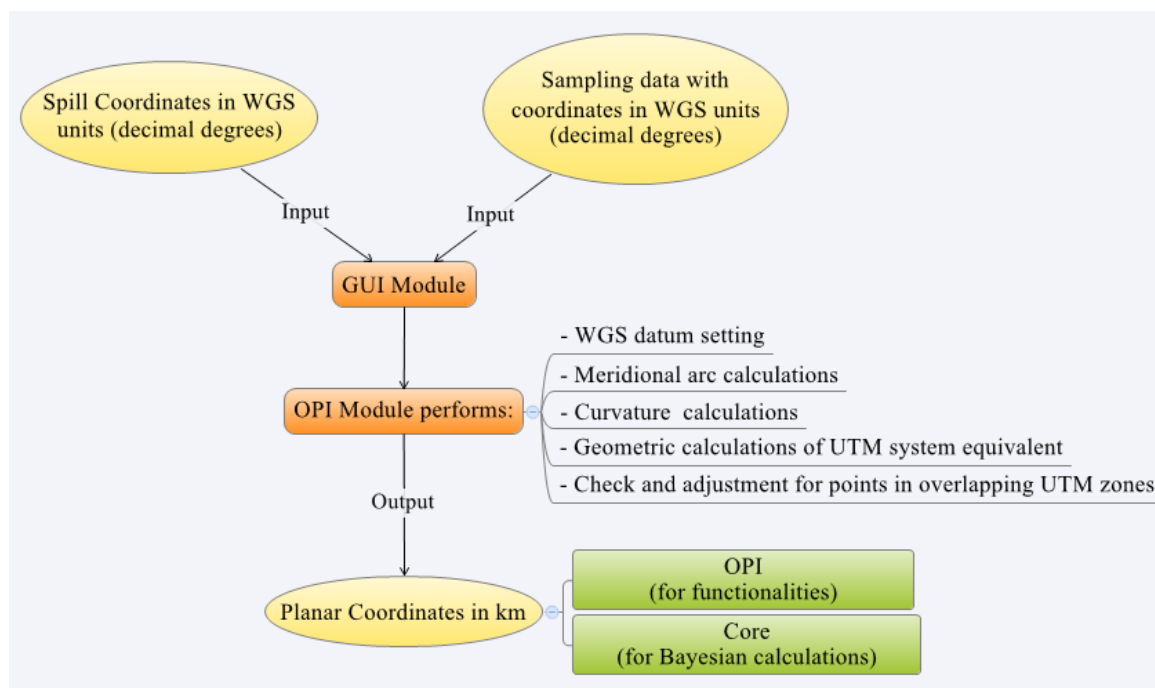


Figure 5.2. Algorithm for conversion of coordinate systems.

Sampling campaigns should be recorded and accepted by the model in World Geodesic System (WGS) units (degrees, minutes and seconds expressed in decimal degrees in the GUI prompts). Interaction of the user with the canvas of the graphic user interface also results in operations in WGS units, because none of the embedded maps are projected to a flat surface. Equations 3.4 to 3.8, on the other hand, all require distances in planar units (SI or metric system). Therefore, conversions using the Universal Transverse Mercator (UTM) projection are used to translate WGS input into distances in kilometers that can be used in the diffusion, advection, and statistical equations, and to convert

gridded model output back to WGS format for graphical presentation in map format. The process, other than involving the creation of global Python functions for the conversions, engages certain computational algorithms to guarantee that (a) precision is not lost in the double conversion process (error is on the order of $1e-9$), and (b) operations over points located at overlapping borders of UTM zones are correctly converted without affecting precision and guaranteeing that user inquiries concerning the locations of sunken oil are accurately displayed on the map canvas of the GUI application.

5.3.3 SOSim Core: Combinatorial Algorithm for the Computation of Multi-Modal, Highly Dimensional Likelihood Functions, Posteriors, Conditionals, and Unconditional Probability Functions

This section describes a numerical method developed to solve a stochastic integral equation that cannot be solved analytically or by Markov chain Monte Carlo simulation due to its high dimensionality. SOSim, the Sunken Oil Simulator requires such a numerical method that is capable of analyzing and using complete proposed parameters domains towards finding the most likely combination of the unknown parameters to predict a state in time, grounded on a statistical description of subjective and limited data. The state in time that needs to be predicted corresponds to the location of sunken oil in relatively flat ocean beds and its transport in time, which follows a set of Bayesian analytical expressions developed in this research and explained in a different chapter.

The analytical model, as seen from Equation (3.1), includes 6 unknown parameters per each predicted source (up to four included), that is, a total of 23 unknown parameters (refer to Chapter 3 for explanation) after applying superposition of sources. Equation (3.1) corresponds to a stochastic calculus integral equation:

$$f(x, y, t) = \int \left(\sum_{j=1}^J \gamma_j f_j(x, y, t | \mu_{x,j}, \mu_{y,j}, \sigma_{x,j}, \sigma_{y,j}, \rho_j) \right) L(C_i | \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\rho}, \boldsymbol{\gamma}) \partial \boldsymbol{\mu} \partial \boldsymbol{\sigma} \partial \boldsymbol{\rho} \partial \boldsymbol{\gamma} \quad (5.1)$$

This model cannot be solved analytically and must be solved numerically. One known solution methodology for solving stochastic integrals are the Markov Chain Monte Carlo (MCMC) processes, which have been proven to work accurately for several parameters, around 5 or 6 in amount. As Equation (3.1) embraces 23 parameters, the dimensionality of the problem did not allow for MCMC use. A matrix approach was also attempted without success because such a highly-dimensional structure was beyond Python's built-in memory and processing capabilities. A combinatorial method that would allow for inspection and use of the complete domain of every unknown parameter towards calculation of functions, and that would have the capability of finding the most likely fields of the domains, was the target of the development of this new method. As a consequence, functions, and ultimately the stochastic integral, were solved using the most likely set among possible combinations of unknown parameters. Such were arranged in data structures characterized by their simplicity in terms of programmable structure, storing, and recovery for further use.

Bayesian computations are distributed throughout three Python classes located in the core module of SOSim. Each class, by definition of object-oriented programming, contains methods. As stated before, the term *methods* refers to functions that perform unique activities to define and to provide different actions to its class, to be used by an extra parameter which is the object that it is to run on, and which belongs within the same class.

Initializing the Bayesian Parameters

The first class of SOSimCore, called *Preliminars*, involves calculating and passing initial setup, global constants and global variables to the other two classes, named *LF* and *Model*, whenever needed. Initial setup includes, for each of the four patches: an assumed integration domain for the coefficient of diffusion, \mathbf{D} , and velocity, \mathbf{v} , vectors; the domain of the correlation coefficient, ρ , from $-0.999 - 0.999$; and the domain of and restrictions on the weighting parameter, γ . Global constants include the number of discretization elements, w , of the domain of the parameters (the same number for all parameters) and the magnitude of the partial differential elements, $\partial\mathbf{D}, \partial\mathbf{v}, \partial\rho$ and $\partial\gamma$, based on w . Global variables are directly related to the programming structure of the Bayesian algorithm, and are produced within the *Preliminars* class using the *doAll()* method. Note that in order to facilitate reference to the parameters in this report, parameters have been divided in two types: the Gaussian parameters and weighting parameters. The first group includes μ_j, σ_j and ρ_j , a total of 5 parameters per patch and

20 overall for four possible patches. In the second group there are the weighting parameters γ_j , one per patch. The weighting parameters each have the same domain, $\{0, 1\}$, but a shared and conditional codomain, $\sum_{j=1}^J \gamma_j = 1$, so that one of the four weighting parameters is not variable but is fixed by the values of the remaining three.

The *doAll()* method starts by finding all possible quadruplets of the weighting parameter, γ , of each patch; that is, by placing all quadruplets of weighting parameters $\gamma_1, \gamma_2, \gamma_3$ and γ_4 that sum to one in a global variable *GammaPossible* of type Python list. This operation is performed accounting for the order of the patches. Thus, further combinatorial operations involving only the possible values of weighting parameters will have the same sorting and can therefore be related. Other important global variables defined in the *doAll()* method are: the number of valid weighting parameter combinations that sum to one (*valid*); the total number of positions (*size*) that would result in a hypothetical array of 5 dimensions (one per Gaussian parameter in a single patch) in which each dimension is discretized equally (using the discretization constant w); and the number of possible sums of bivariate Gaussian combinations times weighting parameter values among patches (*newsze*), that would honor the conservation of mass restriction on the weighting parameters.

Calibrating the Bayesian Model: Calculation of the Likelihood Function

The second class of the core module of SOSim is named *LF*, standing for Likelihood Function. This class executes all operations needed to compute Equation (3.3):

$$L(\mathbf{C}_i | \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\rho}, \boldsymbol{\gamma}) = \prod_{i=1}^I \lambda \exp(-\lambda \mathbf{C}_i), \quad (3.3)$$

in which $1/\lambda = \sum_{j=1}^J \gamma_j f_j(x_i, y_i, t_i | \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, \rho_j)$ and $f_j(x_i, y_i, t_i | \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, \rho_j)$ is the bivariate Gaussian probability function modified by a variable number of Gaussian terms representing superimposed reflected images in the case that a sampling campaign occurs in the nearshore environment. Expanding the function for open-ocean conditions,

$$f_j(\mathbf{X}_i, t_s | \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, \rho_j) = \mathcal{N}(\mathbf{X}_i, t_s | \theta_j) = \frac{1}{2\pi\sigma_{x,j}\sigma_{y,j}\sqrt{1-\rho_j^2}} \exp\left[\frac{-B_{ij}}{2(1-\rho_j^2)}\right], \quad (3.4)$$

$$\text{in which } B_i = \frac{(x_i - x_0 - \mu_x)^2}{\sigma_x^2} - \frac{2\rho(x_i - x_0 - \mu_x)(y_i - y_0 - \mu_y)}{\sigma_x\sigma_y} + \frac{(y_i - y_0 - \mu_y)^2}{\sigma_y^2}.$$

The initialization arguments of the *LF* class include input from the *Preliminars* class (parameters settings and variables) and input from the GUI through the OPI module (processed spill coordinates, x_i, y_i , and sampling campaigns, s , in terms of processed longitudes, latitudes, concentrations, and sampling times, t_s). Equation 3.3 is solved using the *LF* class' method *calculateLV*. The first step in solving this equation is to find

λ at each sampled point i , by finding, for each sampling point, the inverse of the sum of the bivariate Gaussian distributions of all patches, maintaining complete and balanced sunken oil mass within their collective volume.

In order to evaluate the complete parameter space at each sampling point, the concept of a hypothetical array of 5 dimensions, one for each Gaussian parameter within a single patch, was adopted, in which each dimension takes $u = w+1$ discretized values representing the endpoints of the w discretizations of the parameter range. Now, proceeding conceptually to fill all positions within the array with the corresponding result of the bivariate Gaussian distribution $f_j(x_i, y_i, t_s | \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, \rho_j)$, it can be seen that the values of $f_j(x_i, y_i, t_s | \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, \rho_j)$ at time t_i change from one sampling point, (x_i, y_i) , to another. In the *calculateLV* method code, since a matrix analysis of this dimensionality is not possible in Python, a set of nested loops finds and annotates to disk the $f_j(x_i, y_i, t_s | \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, \rho_j)$ value for each and all *positions* within the hypothetical array, that is for each sampling point at every sampling time. Thus, each annotation of $f_j(x_i, y_i, t_s | \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, \rho_j)$ becomes an element of a Python list object, which is stored on disk to reduce memory footprint.

To this point, the description has been of an array corresponding to the first patch. However, it can be seen that because the possible domain of the Gaussian parameters is the same for all four patches, the array described (stored in a Python list object) is the same, such that it contains the same values, for all patches. Once the value of

$f_j(x_i, y_i, t_s | \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, \rho_j)$ at all positions is recorded, the superposition of patches is computed using the four identical 5-D conjectural arrays (actually, Python list objects, hypothetical structures with all values positioned in a list on file). Referring to the set of values of $f_j(x_i, y_i, t_s | \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, \rho_j)$ at each position, for the j -th patch, as \mathcal{N}_j , superposition consists mathematically of summing $\gamma_1 \mathcal{N}_1 + \gamma_2 \mathcal{N}_2 + \gamma_3 \mathcal{N}_3 + \gamma_4 \mathcal{N}_4$, in all possible ways (to use the entire domain of the weighting parameters γ_j) such that weighting parameters sum to unity, $\sum_{j=1}^4 \gamma_j = 1$. Because the 5-D hypothetical array is the same for every patch, $\mathcal{N}_{k,j} = \mathcal{N}_{k,1} = \mathcal{N}_{k,2} = \mathcal{N}_{k,3} = \mathcal{N}_{k,4}$, in which the subscript k , $1 \leq k \leq K = u^5$, is a position index indicating a sampling point and time and a point in the uncertain parameter space.

Figure 5.3 illustrates the computation of superposition using all possible combinations of $\mathcal{N}_{k,j}$ values, for γ_j . As shown, the X-axis contains weighting parameters, γ_j , discretized in steps of 0.1. The Y-axis represents the patches, $j = 1 - 4$. The Z-axis represents the values $\mathcal{N}_{k,j}$. The line traced in the Y-Z plane joins one parameter vector-sample (position), k , of a patch with another one, representing just one of the possible combinations of $\mathcal{N}_{k,j}$. The line traced in the X-Y plane represents one possible combination of values of γ_j . The superposition combination in this figure would be the product of the $\mathcal{N}_{k,j}$ values indicated by a star for the first patch, plus the product

of the $\mathcal{N}_{k,j}$ values indicated by a star for the second patch, and so on for the remaining two patches. Exploration of all the possible combinations and the summations is performed in the code using the *GammaPossible* variable array and the *validList()* generator.

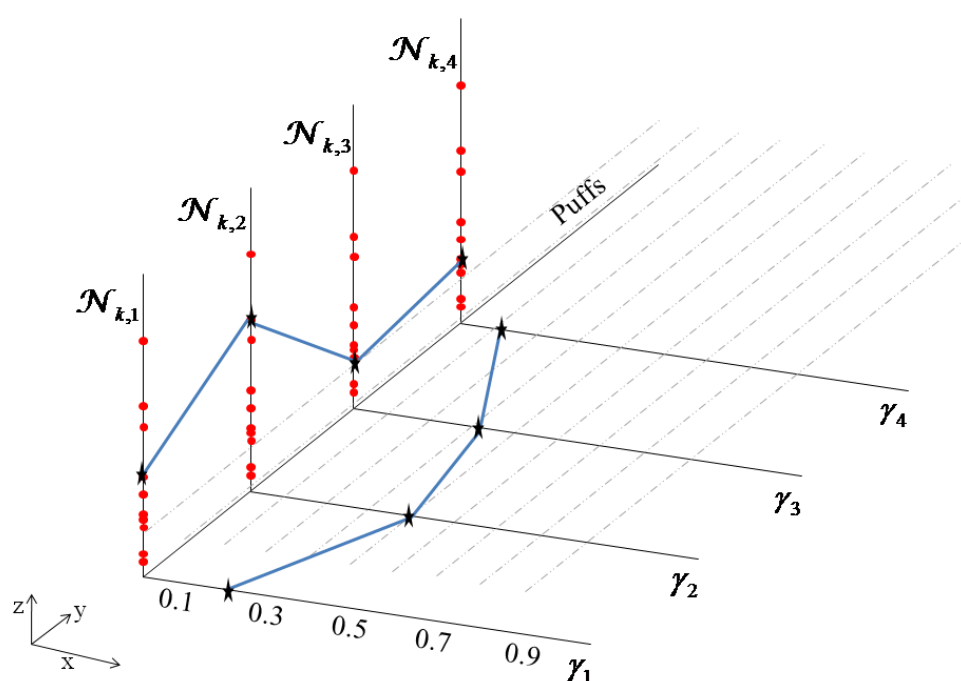


Figure 5.3. Representation of the superposition concept in three dimensions.

GammaPossible is one of the variables produced by the *Preliminars* class under the *doAll()* method. In its initialized form, *GammaPossible* is an empty list. One sub-list is added at a time when four counters traveling through four zero-to-one γ domains

encounter four fractions that sum to one, e.g.: a sub list [0.2, 0.4, 0.3, 0.1] is added to the *GammaPossible* list for the example shown in Figure 5.3. All possible combinations that comply with $\sum_{j=1}^4 \gamma_j = 1$ are appended to *GammaPossible* in this fashion. Components to assemble the superposition list, representing $1/\lambda$ in Equation (3.3), are completed with the *GammaPossible* list, but require an organized one-by-one generation that is attained in the next step.

A generator is a Python structure comparable to a function: the generator, instead of returning a final, condensed variable, has the capability of ‘yielding’ one individual result of a loop calculation at a time, at the point requested by the developer, until the series is exhausted. Generators are resource-efficient and add versatility to the code while reducing memory footprint and, in most cases, run time. The generator *validList()* yields individual superposition values as long as the iterating loop has not been exhausted. The computational methodology for the generator includes looping over the Y-Z plane of Figure 5.3 (*sumComb* argument in the code) to find possible combinations of $\mathcal{N}_{k,j}$, and coupling them with each of the *GammaPossible* sublists that are equivalent to all possible combinations in the X-Y plane of the same figure. Individual superposition yielding for each weighting parameter combination z in *GammaPossible* is called later in the process of building λ . The total number of elements yielded by the *validList()* generator is $Z(u^5)$, in which Z is the number of ways of combining the four weighting parameters γ_j

such that $\sum_{j=1}^4 \gamma_j = 1$, and the exponent five accounts for the number of Gaussian parameters per patch. The *validList()* generator is staged, meaning that it has the capability to divide the total number of elements yielded into different Z groups of u^5 yieldings each, depending on the particular combination z being analyzed.

Continuing in the *calculateLV* method of the *LF* class, the components needed to compute Equation 3.3 are now available. A local variable *supValue* is set equal to each superposition result using a *validList()* generator object, to compute likelihood values for each z , according to Equation 3.3. The precision limits in Python are set at approximately 10^{-300} and 10^{300} , and so variables holding superposition results of such orders of magnitude must be avoided to prevent numerical error. If *supValue* $> 1e-300$, then a likelihood value $LVComb_{k,z,i}$ is calculated as

$$LVComb_{k,z,i} = \frac{1}{supValue_{k,z,i}} \exp\left[-\frac{1.0}{supValue_{k,z,i}} C_i\right] \text{ for every } i\text{-th sampling point, } z\text{-th}$$

combination, and k -th possible element within. If $LVComb_{k,z,i}$ is different from zero, this k^{th} term is included as a term in the sum for the likelihood function of the currently analyzed combination, z , and if $LVComb_{k,z,i}$ is equal to zero, it is discarded. The

summation $\sum_{i=1}^I \ln(LVComb_{k,z,i})$ for each z is performed across sampling points, i , not

across the k elements of the array according to the definition of a likelihood function.

This process leaves a sum for each z , which is still a 1-D array with k positions. Note

$$\text{that } \sum_{i=1}^I \ln(LVComb_{k,z,i}) = \prod_{i=1}^I LVComb_{k,z,i}.$$

To compute this sum in code, the variable *Likelihood* is of type 2-D array, with Z positions in the first dimension and k positions per z element. The *Likelihood* variable can also be understood as one K -element, 1-D array per each possible weighing parameter combination z . While the loop for a point i is being performed, each value in the k position is added to the natural log of the $LVComb_{k,z}$ value that is being produced for the current sampling point. This procedure is repeated until all sampling points in all campaigns are exhausted. As a result, the likelihood function remains a 2-D array with K elements per each z combination, each position inheriting the combinatorial nature of its immediate predecessor, $LVComb$. The last set of instructions performed by the *calculateLV* method occurs outside of the looping structure because the information from all the sampling points and sampling times is already contained in the *Likelihood* local variable. The maximum among the z -dimensions of the 2-D array is computed and the selected maximum (a 1-D array with K elements that belong to the most likely combination, z , of weighting parameters, γ_j), becomes the natural log of the final likelihood function. The preceding maximum likelihood of γ_j calculation allows for the knowledge of the most likely values of the weighting parameters that have been selected to be employed in successive computations. Then for every position, k , in *Likelihood*, the numerical value is transformed back to the linear domain as $Likelihood_k =$

$\exp(Likelihood_k)$. The *Likelihood* array is then the likelihood function. The *LF* class accomplishes these tasks and frees memory by writing the *Likelihood* array to disk for further use.

Predicting: Calculation of the Conditional Gaussian Function

The third Python class of the core module of the software developed to perform the Bayesian calculations is the *Model* class. The main objective of the *Model* class is to complete the computation of Equation 3.1, by computing the conditional bivariate Gaussian distribution, multiplying this result by the likelihood function, and integrating over the parameter space. The latter is found in the code of the method *calculateCG()*. However, the *Model* class includes many other Python methods that share responsibilities with the *SOSimOPI.py* module, such as assigning boundary conditions to the Gaussian equations, managing coordinate systems (WGS v. UTM), and managing model spatial resolution, units and mapping.

The *calculateCG()* method uses a programming approach similar to that described for computing the bivariate Gaussian distribution values and superposition for the likelihood function. The principal difference is that the conditional distribution is computed at the nodes, \mathbf{X}_m , of a regular grid defined for the region of the spill, and user-specified times, t , of prediction, instead of at the locations, (x_i, y_i) , and times, t_s , of field sample collection. The arguments passed to the *calculateCG()* method include grid

coordinates, prediction times and a Boolean variable indicating the presence of land within the modeling area.

Consider again the hypothetical 5-D array (one dimension for each parameter of a single Gaussian patch), in which each dimension has u discretized values. Again, the parameter space is evaluated over the bivariate Gaussian function, written for this case as $f_j(\mathbf{X}_m, t | \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, \rho_j)$. The function is evaluated using the same nested programming structure as described for the likelihood function, with each position written to a Python list object stored on disk. Again, because the same parameter domain is used for all four patches, the new Python list object is identical for all patches. After the value of $f_j(\mathbf{X}_m, t | \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, \rho_j)$, modified or not by the boundary conditions, is written to the list for all positions, superposition is performed as before, with the difference that the most likely combination of weighting parameters has already been selected by the likelihood function using the data. That is, letting $f_j(\mathbf{X}_m, t | \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, \rho_j) = \mathcal{CN}_j$, the sum $\gamma_1 \mathcal{CN}_1 + \gamma_2 \mathcal{CN}_2 + \gamma_3 \mathcal{CN}_3 + \gamma_4 \mathcal{CN}_4$ is computed subject to $\sum_{j=1}^4 \gamma_j = 1$, which has been already accomplished. The procedure depicted in Figure 5.3 is used to find the 1-D array of superimposed values, $CG_{k,m}$, each value representing a possible parameter vector, for each node, (x_m, y_m) , of the user-defined modeling grid. The number of possible elements is now the total number of elements yielded by the *validList()* generator when it uses the \mathcal{CN}_j values as one of its arguments and the most likely weighting parameter

combination, z , as another argument (instead of the complete *GammaPossible* list of possible combinations).

Integrating

The $CG_{k,m}$ arrays are used directly in the integration, according to Equation (3.1), of the product of the conditional Gaussian sampling distribution by the Bayesian posterior distribution, the latter of which is equivalent to the likelihood function. The likelihood function, LF_k , consists of a single array of K elements, one element for each possible parameter vector. The conditional Gaussian distribution, on the other hand, consists of one K -element array, $CG_{k,m}$, for each point, \mathbf{X}_m . The conditional Gaussian array at each point, \mathbf{X}_m , is multiplied on a one-to-one element basis with the likelihood function, as $LF_k \times CG_{k,m}$. The resulting K -element array is multiplied by the product of the Gaussian parameters' differentials and summed at each point, \mathbf{X}_m , as $SOM_m = \left[\sum_{k=1}^K LF_k \times CG_{k,m} \right] \times delta$, in which SOM_m stands for the relative probability of finding sunken oil mass at \mathbf{X}_m , K is the total number of elements $Z(u^5)$, and $delta$ is the product of the Gaussian parameters' differentials. Subsequently, the method *mapping()* of the class *Model* rearranges results to perform plotting, and creates the corresponding input for the mapping methods of the OPI and GUI modules.

The procedure of the *LF* class is repeated always that the user requests a calibration based on data input. The process that the *Model* class performs, on the other hand, uses any existing calibration file or one expressly created and is executed when a prediction in time is bidden by the user. The *Preliminars* class always runs before either the *LF* or the *Model* classes.

Mapping

The *Model* class includes one last method called *Mapping()*. This method does not map by itself, but it rearranges all saved-to-disk raw SOM_m time variables into numerical arrays and creates portable files that can later be processed by the georeferencer unit of the OPI module. At a later stage, the results will be processed to be displayed by the GUI and be modified as the user wishes by use of the post-processing layout of the same module.

5.4 Software Operation

The details of SOSim operation are described in the Users Manual, another product of this research (Appendix A). Characteristics of operation and interpretation of results include:

1. If sunken oil mass is predicted to extend beyond the selected graphical domain for the prediction times requested, maps may be developed from partial re-runs for contiguous areas;

2. Resolution of prediction within any desired area of a plot can be enhanced by running the model for a “zoomed-in” area, following each model run; and
3. Color hues are not constant from figure to figure, but show relative probabilities from high to low at each time. This also implies that a result generated by zooming into another plot will show independent hotspots, not constant in color with respect to the parent figure.

5.4.1 Binary and Source Packages and Support Documentation

The SOSim set of modules along with all its dependencies were consolidated in a distributable, executable Windows 32 package named SOSim 1.0rc1 (release candidate). This first release candidate is expected to work with any Microsoft Windows 32-bit operating system without the need of previous Python installation, or by the installation and use of the source package, as is explained in the User Manual, appended and created as part of this dissertation. The electronic version of the Users Manual contains an electronic demo that supplements the manuscript. The User Manual guides the user through the installation and operation of SOSim. Knowledge of statistics is not required. Input of information as required for modeling a spill, including accident coordinates, sampling data (basic Microsoft Excel® skills are required to assemble a sampling campaign file), and criteria to define desired prediction times. The Users Manual and Demo are distributed with the binary file SOSim.exe.

5.5 Discussion and Importance of Results

A complete programming environment has been developed to allow interactive computation and geographic representation of unconditional probabilities by integration over a highly dimensional uncertain parameter space. The new methodology, in turn, is the core tool for the Bayesian simulation of sunken oil occurrences in relatively flat bottoms of the ocean after sudden spills.

The SOSim predictive Bayesian statistical software stands as a decision tool that is expected to be useful for identifying sunken oil hotspots, addressing the need for tracking of sunken oil following a spill, and to target cleanup activities and to support cleanup termination decisions using a minimum of resources from responder teams.

Results are displayed in interactive maps of geographic areas, produced based on user input supplied to an interactive graphical user interface. The model was developed with the intent of being used on desktop computers for maximum accessibility, and therefore the intrinsic computational limits and approaches incorporated into the model are consistent with this vision. However its performance can be enhanced considerably if this condition is relaxed.

CONCLUSIONS

A novel statistical data-limited technique representing a cross between a statistical static sampling plan and a contaminant transport model is developed conceptually, analytically, and numerically in this thesis. Specific achievements of the development include the derivation of unconditional probabilities of relative pollutant mass as a function of time and space; the development of a method of computing Lagrangian relative concentration profiles subject to variable, continuous or discontinuous, irregular, approximately curved boundaries in two dimensions applied to Gaussian-distributed sources; a combinatorial algorithm for the computation of multi-modal, highly dimensional stochastic functions, and the successful assembly of a software package to house the model, including a graphical user interface and a processing and operative module that allows for stand-alone model execution. The complete tool developed in this research was named the Sunken Oil Simulator – SOSim–, and is expected to be used as a decision-making tool for agencies, planners, and emergency responders of oil spills.

During the testing and confirmation cases presented in this dissertation, the model was shown able to utilize irregularly-sampled, limited available field data collected shortly after a spill event and recreate the data sets at their time of occurrence by the use of the predictive technique employed. The model also has the checked capability of updating based on additional, relative synthetic data that modifies the calibration files (understood as the likelihood function), from possibly different and irregular geographical areas, as they become available. The model is able to present unconditional

probabilities of sunken oil in output maps that belong each to a user-requested prediction time. The model accounts for the time lapse before depositing on the bottom that oil may experiment due to its potential for sinking and short-term weathering. The software has also demonstrated multimodality capability, which is the aptitude of the predictive model to infer from the data whether or not the sunken oil is distributed in single or multiple patches, and to track and predict this multimodal behavior in time, as possibly tested. Finally, the model has the functionality of inferring oil location in time based on subsequent synthetic data together with shoreline boundaries and the values or estimated ranges of values of the coefficients of advection and diffusion, and this works according with the expectations; however, the model itself is not considered verified in real life since one more real data set was necessary for comparison of results obtained in time. Nevertheless, the functionalities intended to arm the model with have been confirmed and equifinality based on a maximum-likelihood assessment of the parameters has been shown. The Sunken Oil Simulator is open to continued verification as this is recommended for refinement.

The new variation of the method of images to provide predictive Gaussian-distributed sources with variable, continuous or discontinuous, irregular, approximately curved boundary conditions assumed to occur nearby the coastal zone has been successful. The new method is considered an innovation, especially for the field of stochastic modeling, since the effect of boundary conditions can be modeled even under the restrictions implied by the lack of known parameters. In addition, the use of computer programs

eases the exercise of intricate geometries generalized in numerical algorithms, opening the possibility of modeling complicated curved boundary conditions as desired.

A complete programming environment has been developed in order to make possible the functioning of a new methodology created for the numerical analysis and solution of a stochastic analytical model. The new methodology, in turn, is the core tool for the Bayesian simulation of sunken oil occurrences in relatively flat bottoms of the ocean after sudden spills. Accurate operation of the functionalities of the model, demonstrated in Chapter 3 of this dissertation, suggest that the combinatorial methodology is applicable to probabilistic contaminant transport modeling when collectively established along interfacing modules that feed it correctly and later retain its results for processing, arranging and proper displaying.

The SOSim predictive Bayesian statistical software stands as a decision tool that is expected to be useful for identifying sunken oil hotspots, addressing the need for tracking of sunken oil following a spill, and to target cleanup activities and to support cleanup termination decisions using a minimum of resources from responder teams. The model and the software represent a new approach to pollutant tracking by inference from limited field data alone. It also poses a novel mapping of unconditional relative probabilities of finding sunken pollutant mass in space and time with rigorous accounting of uncertainty. The model was developed in the open-source Python programming language, for potential interface with new response, cleanup, and damage assessment models.

Limitations of the model are principally in terms of the types of information that can be exploited as model input, and model resolution. Model refinements and enhancements are suggested, including the addition of the capability of predicting sunken oil locations in non-flat ocean bottoms, that is, the inclusion of bathymetry as prior information; also, bottom current information is becoming available, e.g. for the Gulf of Mexico which supports high density oil vessel traffic and drilling operations, and capability for entering such information as prior information would be recommended. Other prior information may be related to such factors as oil type and geographical environment of the spill, factors that may imply the possible development of Bayesian prior probability distributions for coefficients of advection and dispersion in directions perpendicular and parallel to shore. Finally, given adequate processing power, several aspects of model resolution could be enhanced, including resolution of the uncertain parameter space to be integrated over, resolution of the geographical area to be modeled, the number of vertices allowed to approximate the shoreline, and the number of different polyline boundaries to account for separated landforms such islands and straits. With additional processing power, it may be possible to implement the model in three dimensions, for application to oil suspended in the water column.

WORKS CITED

- Aitchison, J. and I. Dunsmore. 1975. *Statistical Prediction Analysis*. Cambridge University Press, New York, Chapter 2, 1975.
- Anex, R. and J. Englehardt. 2001. Application of a Predictive Bayesian Model to Environmental Accounting. *Journal of Hazardous Materials*, Elsevier, vol. 82, no. 2, pp. 99-112.
- Barker, C. 2009. Personal electronic communication. Oceanographer, National Oceanographic and Atmospheric Administration, NOAA Ocean Service Office of Response and Restoration, October 20, 2009.
- Bear, J. 1979 *Hydraulics of Groundwater*. McGraw-Hill, 1979.
- Beegle-Krause, CJ, C. Barker, G. Watabayashi and W. Lehr. 2006. Long-Term Transport of the Oil from the T/B DBL-152: Lessons Learned for Oils Heavier than Seawater. Arctic and Marine Oil Pollution Conference 2006 Proceedings, Vancouver, B.C., Canada, June 6-8, 2006. Ottawa, Ontario: Environment Canada.
- Beegle-Krause, CJ. 2001. General NOAA Oil Modeling Environment (GNOME): A Neww Spill Trajectory Model. Hazardous Materials Assessment Division, Office of Response and Restoration, National Ocean Service, NOAA. Seattle, Washington, 2001.
- Bloetscher, F., J. Englehardt, D. Chin, J. Rose, G. Tchobanoglous, V. Amy, and S. Gokgoz. 2005. Comparative Assessment of Municipal Wastewater Disposal Methods in Southeast Florida. *Water Environment Research*, vol. 77, no. 5, pp.480-490.
- Boehm, P.D., Barak, J.E., Fiest, D.L. and Elskus, A.A. 1982. A Chemical Investigation of Transport and Fate of Petroleum Hydrocarbons in Littoral and Benthic Environments: The Tsesis Oil Spill. *Marine Environmental Research*, v6, 1982, pp. 157-188. Massachusetts, USA.
- Cedre. 2007. *Understanding Black Tides*. Centre of Documentation, Research and Experimentation on Accidental Water Pollution, Brest Cedex, France. Website accessed 2010. <http://www.cedre.fr/en/contact.php>.
- Chin, D.A. 2006. *Water-Quality Engineering in Natural Systems*, Wiley, 2006.

- CRRC. 2007. Coastal Response Research Center. FY 2007 Annual Request for Proposals. May 16, 2007
- CRRC. 2007. Coastal Response Research Center. Submerged Oil – State of the Practice and Research Needs Workshop. Coastal Response Research Center, Durham, New Hampshire, December 12-13, 2006, Draft Report Issued May, 2007.
- Dean, R.G., and R.A. Dalrymple. 1991. Water Wave Mechanics For Engineers and Scientists. World Scientific, Singapore, 1991.
- Douligeris, C., E. Iakovou, J. Englehardt, H. Li, C. Ip, and C. Mooers. 1998. Development of a National Marine Oil Transportation System Model. Spill Science & Technology Bulletin, Elsevier Science Ltd., vol. 4, no. 2, pp. 113-121.
- Einstein, Albert; R. Fürth, transl. by A. D. Cowper. 1926, reprinted in 1956. On the Movement of Small Particles Suspended in a Stationary Liquid Demanded by the Molecular-Kinetic Theory of Heat. Investigations on the theory of the Brownian motion. Dover Publications, 1956.
- Englehardt, J. 2004. Predictive Bayesian Dose-Response Assessment for Appraising Absolute Health Risk from Available Information. Human and Ecological Risk Assessment, The Association for Environmental Health and Sciences, vol. 10, no. 1, pp. 69-74.
- Englehardt, J. and J. Swartout. 2004. Predictive Population Dose-Response Assessment for Cryptosporidium Parvum: Infection Endpoint. Journal of Toxicology and Environmental Health Part A: Current Issues, vol. 67, no. 8-10, pp. 651-666.
- Englehardt, J., H. An, H., L. Fleming, and J. Bean. 2003. Analytical Predictive Bayesian Assessment of Occupational Injury Risks: Municipal Solid Waste Collectors. Risk Analysis, Society for Risk Analysis, vol. 23, no. 5, pp. 919-927.
- Englehardt, J., and C. Peng. 1996. A Bayesian Benefit-Risk Model Applied to the South Florida Building Code. Risk Analysis, Society for Risk Analysis, vol. 16, no. 1, pp. 81-92.
- Englehardt, J. 1995. Predicting Incident Size from Limited Information. Journal of Environmental Engineering, American Society of Civil Engineers, vol. 121, no. 6, pp. 455-464.

- Free Software Foundation, Inc. 2007. GNU General Public License, version 3, June 2007. Site accessed May 2010. <http://www.gnu.org/licenses/gpl.html>.
- General Counsel for Natural Resources/NOAA Office of Response and Restoration. 2009. Case: Tank Barge BDL 152, Federal Waters of the Gulf of Mexico. Site accessed 2010. <http://www.darrp.noaa.gov/southeast/dbl152/index.html>, revised by GCNR May 2009.
- ITOPF. 2009. The International Tanker Owners Pollution Federation Limited. Website accessed 2010. <http://www.itopf.com/>, 2009.
- Jaynes, E. 1957. Information Theory and Statistical Mechanics. *Physical Review*, 106, 620-630, 1957.
- Kapur, J. 1989. *Maximum Entropy Models in Science and Engineering*. Wiley & Sons, New York, pp. 67-68, 1989.
- NOAA. 2006. Submerged Oil Working Group. Technical Guidelines on Sunken Oil Assessment and Removal Techniques, Website last accessed 2010, http://www.crrc.unh.edu/submerged_oil/guidelines_sunken_oil_italy2.pdf
- Obie, D. and J. Englehardt. 1996. A Bayesian Model to Predict Oil Spill Consequences of Management Plans in the Gulf of Mexico. Proc. Eco-Informa '96: Global Networks for Environmental Information, Epcot Science and Technology, Lake Buena Vista, FL, 4-7 November.
- Ojo, T.O, J. Bonner, and C. Page. 2007. Simulation of Constituent Transport Using Reduced 3D Constituent Transport Model (CTM) Driven by HF Radar Model Application and Error Analysis. *Environmental Modeling and Software*, Elsevier, V. 22, pp 481-501, 2007.
- Python Software Foundation. 2010. About Python, Remarkably Powerful Programming Language. Web site last accessed July 2010. <http://www.python.org/about/>.
- PSF. 2001. Python Software Foundation. Website accessed 2010. <http://www.python.org/>
- QGIS Organization. 2009. Website last accessed April 2010, <http://www.qgis.org>.
- Qt Development Frameworks. 2009. Website last accessed April 2010, <http://qt.nokia.com>.

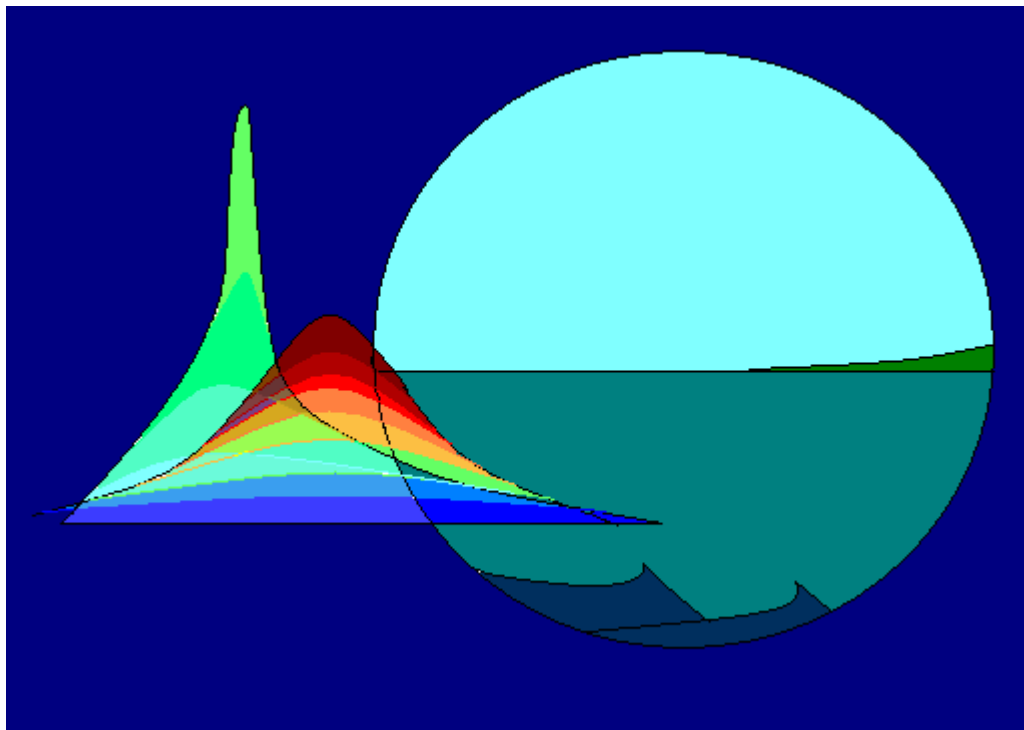
- Research Planning Inc. 2001. Sunken Oil Pathways: Protocols for NRDA Surveys. Website accessed 2010, <http://www.researchplanning.com/services/nrda/Sunken.oil.pathways.pdf>.
- Shannon, C. E. 1948. A Mathematical Theory of Communication. The Bell System Technical J. 27, 379-423 and 623-656, July and October 1948.
- Spaulding, M.L., T. Opishinski and S. Haynes. 1996. COASTMAP: An Integrated Monitoring and Modeling System to Support Oil Spill Response. Spill Science and Technology Bulletin, Vol. 3, No. 3, pp. 149-169, 1996.
- Spaulding, M.L., V.S. Kolluru, E. Anderson, and E. Howlett. 1994. Application of Three-Dimensional Oil Spill Model (WOSM/OILMAP) to Hindcast the Baer Spill. Spill Science and Technology Bulletin, Vol. 1, No. 1, pp. 23-35, 1994.
- Sugioka, S., T. Kojima, K. Nakata, and F. Horiguchi. 1999. A Numerical Simulation of an Oil Spill in Tokyo Bay. Spill Science and Technology Bulletin, Vol. 5, No. 1, pp 51-61, 1999.
- Summerfield, M. 2007. Rapid GUI Programming with Python and Qt, the Definitive Guide to PyQt Programming. Prentice Hall, 2007.
- Todd, D.K., L. Mays. 2005. Groundwater Hydrology. Third edition, Wiley, 2005.
- United States Coast Guard, U.S. Department of Homeland Security. 2005. Press Release: Update on Rough Weather Hinders Barge Salvage Operations. November 29, 2005. <http://www.piersystem.com/go/doc/425/89575/>, as of April 2010.
- Winkler, R. 2003. Bayesian Inference and Decision, second edition. Probabilistic Publishing, 2003.

APPENDICES

Appendix A: Users Manual

SOSim User Manual

A Guide to Operate the Sunken Oil Mass Simulator



By M. Angelica Echavarria Gregory

In Conjunction with the Dissertation “Development of a Predictive Bayesian Data-Derived Multi-Modal Maximum-Likelihood Gaussian Model for Simulation of Sunken Oil Mass in Time

TABLE OF CONTENTS

	LIST OF FIGURES AND TABLES.....	156
Section		
1	INTRODUCTION.....	157
1.1	What is SOSim?.....	157
1.2	Why Python?.....	159
1.3	Objectives of this User Manual: A Guide to the SOSim GUI.....	161
1.4	Scope of Model Applicability.....	162
2	INSTALLATION.....	164
2.1	Hardware Requirements.....	164
2.2	Software Prerequisites.....	165
2.3	Installation from the Executable File.....	166
2.4	Alternate Installation from Source.....	166
3	INPUT.....	169
3.1	Spill Information.....	170
3.2	Sampling Campaign(s).....	173
3.3	Modeling Area and Grid.....	179
3.4	Land Boundaries.....	182
3.5	Prediction Times.....	184
3.6	Default Input.....	186
4	PROCESSING.....	188
4.1	The Calibrate, Calibrate + Run, and Recalculate Buttons.....	188
4.2	Run Time and Progress Bar.....	189
5	OUTPUT.....	190
5.1	Default Output.....	190
5.2	Optional Output.....	191
6	POST PROCESSING.....	195

7	PORTABILITY OF RESULTS.....	196
8	SOFTWARE PORTABILITY.....	197
	REFERENCES.....	198

LIST OF FIGURES AND TABLES

Figure

3.1	Main screen of the computer application, before starting a new project.....	169
3.2	Spill Information input prompts in the GUI.....	170
3.3	Marked spill site after selection of longitude, latitude values and directions.....	173
3.4	Creation of a sampling campaign file. Recording spatial coordinates and observed relative concentration values (scale 0 to 100) for each sampling point in an Excel file.....	175
3.5	Sampling Campaign input prompts and buttons in the GUI.....	176
3.6	Example uploaded sampling campaign data file.....	178
3.7	Modeling Area and Grid input prompts and buttons in the GUI.....	180
3.8	Selecting the modeling area in the GUI.....	181
3.9	Land Boundaries input buttons and spin box in the GUI.....	182
3.10	Prediction Times input buttons in the GUI.....	184
3.11	Modify default input settings dialog box.....	187
4.1	Processing buttons.....	188
5.1	Pan View button set.....	191

Table

2.1	Software dependencies for SOSim installation from source.....	165
-----	---	-----

1. Introduction

SOSim is a modeling tool developed to help locate sunken oil in relatively flat bays based on limited available field data collected shortly after a spill, when oil has begun appearing on the bottom. This User Manual will guide the operator through model installation to model operation and results management, to obtain maps of relative probabilities of finding sunken oil at user-specified times of prediction that are not conditional upon the values of uncertain parameters of the model.

1.1. What is SOSim?

SOSim is a predictive Bayesian multi-modal Gaussian model of relative probabilities of finding sunken oil at points on a bay bottom and in time, designed to accept primary information in the form of limited field data at one or more sampling times. The predictive relative probabilities produced are not conditional on the values of uncertain model parameters such as the water velocity and coefficients of dispersion on the bay bottom. These probabilities can be interpreted as *relative oil concentrations*, depicted to occur on the bottom in somewhat more disperse patches than are actually occurring due to uncertainty in the advective and dispersive forces acting on the oil at depth. Due to the lack of information on the total oil sinking as a function of time, the model cannot assess absolute concentrations, but rather relative concentrations showing oil “hotspots” and areas where oil may not be collecting. All the functionalities designed for the model have been tested and verified, but a formal verification of the results has not been possible yet

as it depends on deployable data distributed in two sets in time: one to be used as input, and one more to compare results with. It is expected to convert the current release candidate version 1.0rc1 to a 1.0.2 version as soon as the complete data sets are gathered after a spill occurrence.

SOSim has been programmed in Python in its entirety. The SOSim model, although using and relying for much of its functionality on several existing Python packages and modules, consists of three principal Python modules developed by the author of this User Manual: the graphical user interface (GUI) module, the operating and processing interface (OPI) module, and the core module. The GUI module (`ui_SOSim.py` in the electronic source package) automatically layouts and keeps characteristics of widgets, labels, canvases and templates in the graphic user interface, holds raw user's input and imports Windows palettes and display. The OPI module (`SOSimOPI.py` in the electronic source package) is the executable file. It imports and links all other modules together, it captures the input information that is entered by the user in the GUI module and operates interrelated buttons and activities of the GUI; it filters, organizes, and processes the input; passes ready-to-use variables and attributes to the core code; accepts modeling results back from the core module; processes the results; and sends display signals to the GUI module's canvas layout for it to depict relative sunken oil concentrations on a map for further user interaction. It also controls modal behavior of the main windows, pop outs, menus and toolbars. The core module (`SOSimCore.py` in the electronic source package) uses variables and attributes passed by the OPI module to compute the

predictive Bayesian relative concentrations, saves output files, and passes results back to the OPI module for display by the GUI or for further use. The GUI has the capability to communicate with the core module through the OPI module during a model run, after a predictive result has been presented, to allow the user to request the modeling of contiguous or other areas as needed.

The GUI contains three basic layouts other than the main toolbars and menus: (1) the pre-run layout or input section, where the user is prompted for input information concerning the spill, sampling campaign(s) and time(s), land boundary, desired modeling area, and prediction times, (2) the canvas layout, where results are displayed in interactive, georeferenced maps, and (3) the post-run layout or output section, which contains tools that allow the user to display the results at different times of prediction, run the model for contiguous areas, save results, print images, and perform other formatting tasks.

1.2. Why Python?

The Python programming language was chosen for development of the SOSim model based on the following criteria:

- The programming software is non-proprietary or non-commercial and open source,
- The programming environment supports relatively rapid prototyping and testing,

- The language is robust in terms of functioning in various operating environments with minimal damage, alteration or loss of functionality, and
- The source code is viewable at all times.

Python is a general-purpose, high-level programming language first released by Guido Van Rossum in 1991 in The Netherlands. Python is flexible, applicable to many programming domains and available on many platforms. The language has an open, community-based development model managed by the non-profit Python Software Foundation. Python's design philosophy emphasizes programmer productivity and code readability (core syntax and semantics are minimalistic, employing white spaces and easy commands). Python supports multiple programming paradigms (primarily object oriented) and features a fully dynamic type system and automatic memory management, similar to Perl, Ruby, Scheme, and Tcl.

The Python programming language has an open, general public license (GPL) that allows source code modification, addition of original code as "imported" packages, and publication of derived work (as GPL). Packages that are created around the world are available to the Python community. Python has also been used as an extension language for many existing systems, including GIS, Web programming, numerical tools, and the R statistical package.

1.3. Objectives of this User Manual: A Guide to the SOSim GUI

The objective of this user manual is to guide the operator through installation and operation of SOSim. The GUI is the point of connection between the user and the processing modules, which perform all required processing of raw input from users and provide the core module with the appropriate information. Among the most important tasks of the GUI, aided by the OPI module, are:

- Collection of basic information on the spill in terms of time of occurrence, and coordinates (longitude and latitude);
- Display of the location of the spill on the map canvas within a 2-degree radius around the spill to allow selection of the desired modeling area;
- Storage and processing of input data from single or multiple sampling campaigns;
- Assignment of the resolution and scale (rectangular) of the results based on default values or a user-defined number of modeling nodes in each direction: north-south and west-east, within the desired or default modeling area;
- Setting of customized dates and times at which sunken oil mass prediction is desired,
- Processing, storage, and passage to the core module of user input;
- Acceptance of output from the core module and geo-referencing of it on the local map; presentation of relative probabilities of finding sunken oil, (0-1), on a relative, color-coded scale, with hotspots in red (1) fading to blue (0). Although

probabilities of sunken oil decrease in time, the hue is re-rendered in every time calculation, such that the areas with the highest probability of finding sunken oil are shown in red at each modeled time, independently of relative probabilities shown for other times of prediction; and

- Processing of optional post-run operations include display of results at different prediction times and performance of partial recalculations to view results for contiguous modeling areas. Useful documentation functions including saving the results as images and printing are also included in the GUI.

1.4. Scope of Model Applicability

Version 1.0rc1 of SOSim is designed for application to:

- Sunken oil;
- Relatively flat bay bottoms, dredged bays, reef flats and lagoons or pools protected by offshore rocks; bays with steeply sloped bottoms would require capability for the use of bathymetric data as prior information, a possible future enhancement;
- Resolution down to the scale of the tidal excursion (oil locations effectively averaged across this excursion);
- Prediction up to the time that the majority of the oil mass is predicted to reach the shoreline;

- Discrete accidental oil releases (as opposed to natural, progressive oil seepage); and
- Relatively uncomplicated concave and convex shoreline geometries; modeling in straits, inland water bodies, harbors, islet areas, and like geographies are not addressed due to computational limitations and the sometimes transient nature of small-scale features.

2. Installation

2.1. Hardware Requirements

SOSim Version 1.0rc1 has been developed for use within the Microsoft Windows 32 bit operating system environment, though porting to other platforms including the Mac OS and Microsoft Windows 64 bit could be considered for future versions. To achieve reasonable performance in terms of computational speed (hours), a 3.0 GHz processor or better is required. In cases in which the machine will be expected to perform tasks in addition to SOSim, an active duo or second processor is required.

SOSim can run on a computer with a page file (virtual memory) of minimum 2.3 GB. Nevertheless, it is recommended that the memory card is of a minimum of 3.0 GB. Memory requirements of SOSim are determined by the fact that Python can allocate memory only up to a total of 2.3 GB, including memory required for all machine functions prior to running the model, when implemented on the Windows 32 bit platform (this limitation is not expected if the model is developed in the future for the Windows 64 bit OS). The total memory used by all processes before starting to run SOSim is typically about 512 MB on machines not having many applications installed and many idle processes to run by default, except for Windows 7 and some editions of Windows Vista which may consume up to 1 GB when idle. Therefore, for the majority of spill cases to be solved with optimal resolution and including recalculations, it is estimated that a computer would require an available memory of about 1.7 GB (that is, a difference of about 1.7 GB between the 2.3 GB limit and the kernel memory taken up by idle

processes). Indirect warning messages provided by the GUI will guide the user in setting the best possible resolution to achieve optimal performance in terms of memory.

2.2. Software Prerequisites

If installing from the binary package, everything needed is included with the executable file. If installing from source, the required programs and modules, with required installation locations, are given in Table 2.1. Three of these packages (Matplotlib, Dateutil and Pytz) are included in labeled folders in the SOSim distribution along with the source code folder. The three individual folders must be placed in the directory C:\OSGeo4W\apps\Python25\Lib\site-packages once the OSGeo4W package is installed, as shown in the table. In addition, some environment variables, paths, and dynamic link libraries (.DLLs) must be changed/added to the machine or a given module when installing from source, as described the Installation section.

Table 2.1. Software dependencies for SOSim installation from source

Package/Module	Install in folder	Link
OSGeo4W 1.0 Kore for Win32	C	http://trac.osgeo.org/osgeo4w/
Matplotlib v. 0.91.2	C:\OSGeo4W\apps\Python 25\Lib\site-packages	http://pypi.python.org/pypi/matplotlib/0.91.2
Dateutil v. 1.2-mpl	C:\OSGeo4W\apps\Python 25\Lib\site-packages	http://pypi.python.org/pypi/python-dateutil/1.2
Pytz 2007g-mpl	C:\OSGeo4W\apps\Python 25\Lib\site-packages	http://pypi.python.org/pypi/pytz/2007g

2.3. Installation from the Executable File

To install SOSim from the executable file, double-click on the SOSim.exe folder and follow the instructions of the installation wizard.

2.4. Alternate Installation from Source

When using the source code of SOSim, Python must be installed from the OSGeo4W console, as given in Table 2.1. This installation procedure is recommended only for users familiar with Python programming. Steps required are as follows.

1. Install, compile, or copy/paste from the SOSim distribution the software/modules indicated in Table 2.1 in the required locations;
2. Add to your system or edit the environment variables listed below,¹

¹ To change an environment variable in a Windows 32 bit machine, right-click on your “My Computer” icon and select “Properties”. In the System Properties box, under “Advanced”, select “Environment Variables” at the bottom. System variables are in the lower part of the box and you only have access to them if you have administrator privileges over the machine. Screen the list to see of the environment variable that you need already exists; if it does, click on it and on “Edit”, then type a semicolon after the existing text and type the given path following, without spaces. If the environment variable does not exist, click on “New” and type the name in capital letters, then the given path.

PATH: C:\OSGeo4W\apps\qgis;%PATH%

PYTHONPATH: C:\OSGeo4W\apps\qgis\python

QGISHOME: C:\OSGeo4W\apps\qgis

Environment variables must be changed from an account with “administrator” access privileges, and for all users. To edit/add environment variables, go to “System” in the control panel and click on the “Environment Variables” button under the “Advanced” tab. Then manage the “System variables” as instructed.

3. Find the `qgis_core.dll` and `qgis_gui.dll` libraries in `C:\OSGeo4W\apps\qgis\bin`. Copy and paste them to `C:\OSGeo4W\apps\qgis\python\qgis` for their use by the Python distribution from the OSGeo4W console;
4. Copy the folder “SourceCode” that comes with the SOSim package to the folder `C:\OSGeo4W\apps\Python25\Scripts`. Add this location to your system path in Python; and
5. Open the module “SOSimOPI.py” and edit the current module’s “path2” variable to the tree file that directs to your Desktop; if this is not done, graphic results will not be displayed and may not be automatically saved.

Now the OPI together with the GUI and essential source code is callable using the “execfile” Python command from the OSGeo4W console in two ways: (1) from the DOS directory `cd C:\OSGeo4W\apps\Python25\Scripts\SourceCode` by typing `SOSimOPI.py`;

or (2) from any DOS tree location, type: *python*, then *import sys*, then *sys.path.append("C:/OSGeo4W/apps/Python25/Scripts/SourceCode")*, and lastly *execfile("C:\OSGeo4W\apps\Python25\Scripts\SourceCode\SOSimOPI.py")*. The SOSim GUI will then pop up.

3. Input

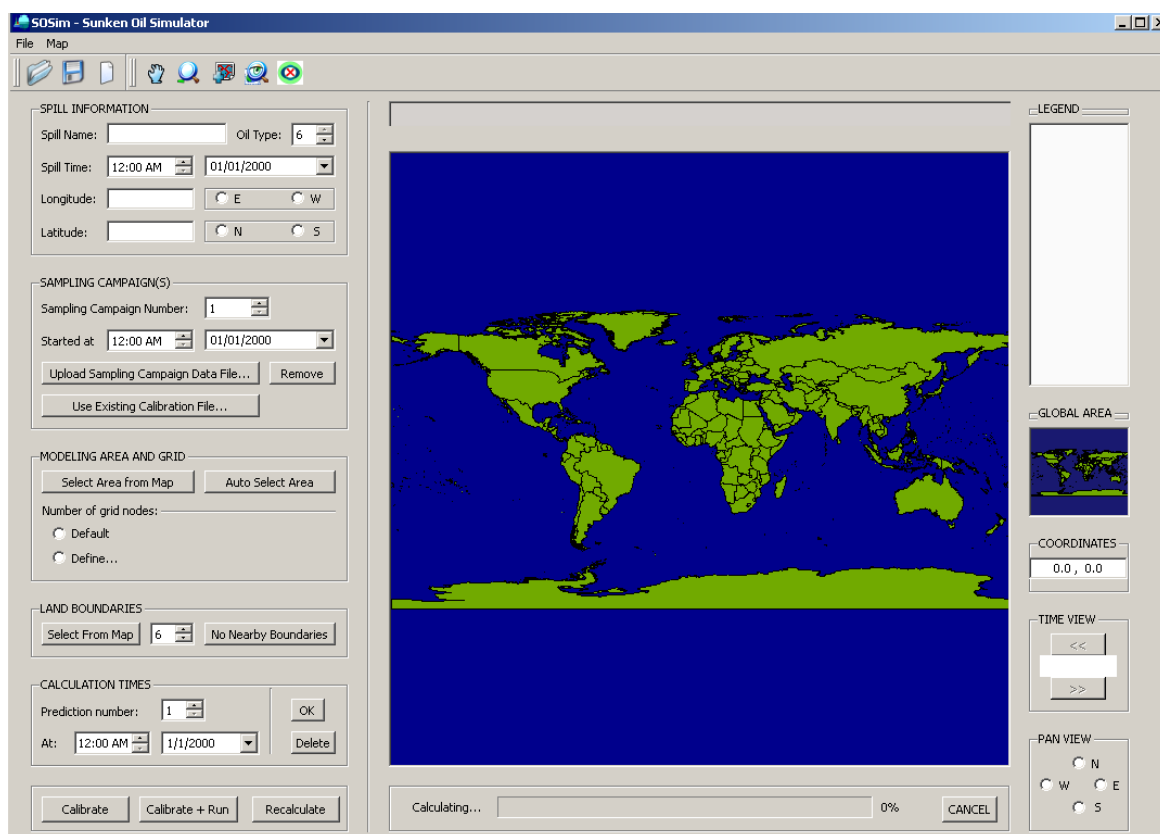


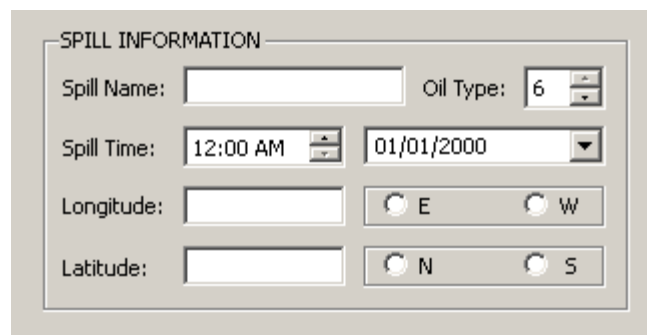
Figure 3.1. Main screen of the computer application, before starting a new project.

Required input is prompted in the area located to the left of the canvas (see Figure 3.1) . The results of the prediction depend on the input. In particular, movement of the oil will be predicted based on relative oil concentrations at the field sampling points at and around a spill site, the time and location of the spill and of the samples collected, and shoreline boundaries. Other essential model input determining output characteristics

including run time and resolution includes basic geographic information about the spill, extent of the modeling area, desired spatial resolution of the graphical output, geographic conditions within the modeling area such as proximity to the shoreline, and the dates at which predictions are desired. The GUI's input area is organized according to functionality, as described in the following sections.

3.1. Spill Information

Information on the spill name, oil type, location, and time are entered as shown in Figure 3.2.



The screenshot shows a window titled "SPILL INFORMATION" with the following fields and controls:

- Spill Name:** A text input field.
- Oil Type:** A dropdown menu currently showing the value "6".
- Spill Time:** A time selection control showing "12:00 AM".
- Date:** A date selection control showing "01/01/2000".
- Longitude:** A text input field followed by two radio buttons labeled "E" and "W".
- Latitude:** A text input field followed by two radio buttons labeled "N" and "S".

Figure 3.2. Spill Information input prompts in the GUI.

This spill information is used to set the geographic system of reference for the run. Required information includes the spill time and the coordinates at which the accident most likely occurred. A spill name is also prompted for use in saving and distinguishing output figures and files. The oil type refers to the classification of oils into six generally

accepted classes by weight and other properties affecting their behavior in the environment (Research Planning Inc, 1994), as explained below under *Oil Type*.

The following conditions apply to the Spill Information input section:

- Only a point source spill occurring at a pair of coordinates, in degrees of longitude and latitude) can be modeled;
- *Spill Name* prompt: allows the user to set the title of the spill, e.g. “DBL-152”. Characters /, \, *, <, >, “, | and ? are not accepted but blanks between words are recognized. If a change in typing is required, an informational message will pop up after all inputs to the Spill Information panel have been entered;
- *Oil Type*: set the spin box from 1 to 6 according to the type of spilled oil, as follows (Research Planning Inc, 1994). Type 1 oils are very light, perhaps ~31 °API gravity, including gasoline and very volatile hydrocarbons. Type 2 are moderately volatile and soluble, including jet fuels, diesel fuel, number 2 fuel oil, and light crude oils. Type 3 includes most crude oils, characterized by their persistence and diminished propensity to evaporate (about one third of the total mass evaporates within 24 hours). Type 4 oils may have ~10 °API gravity, little propensity to evaporate or dissolve, and high likelihood of sinking. Type 5 oils have essentially no evaporation potential, weather very slowly, and sink immediately, including heavy industrial fuel oils. Type 6 oils include heavy animal or plant oils.

- *Spill Time*: set the spin box to the time at which the most significant oil loss occurred. Notice that you can edit the hour and minutes using either the mouse cursor or the arrows of the spin box, or both. Then, set the day, month and year of the spill using the smart calendar that pops up when you click in the drop-down menu located to the right. The date line edit can also be changed manually using the mouse cursor and the keyboard.
- *Longitude*: this line prompts for a decimal quantity in degrees (WGS) corresponding to the longitude coordinate at which the spill occurred. All quantities must be greater than zero and must have decimal figures that can be zero. In front of the prompt the user shall select the radio button “E” (longitude east) or “S” (longitude south), which will assign a direction to the quantity you typed and will guide the canvas to the world’s geographic meridian closest to the longitude of the spill site.
- *Latitude*: this line prompts for a decimal quantity in degrees (WGS) corresponding to the latitude coordinate at which the spill occurred. All quantities must be greater than zero and must have decimal figures that can be zero. In front of the prompt, select the radio button “N” (latitude north) or “S” (latitude south), which will assign a direction to the quantity you typed, will guide the canvas to the world’s geographic parallel closest to the latitude of the spill site, and will mark the spill site with an X (Figure 3.3).

- If you wish to modify the spill name at this point, you would have to proceed with the change and then repeat the last step 'Latitude'.

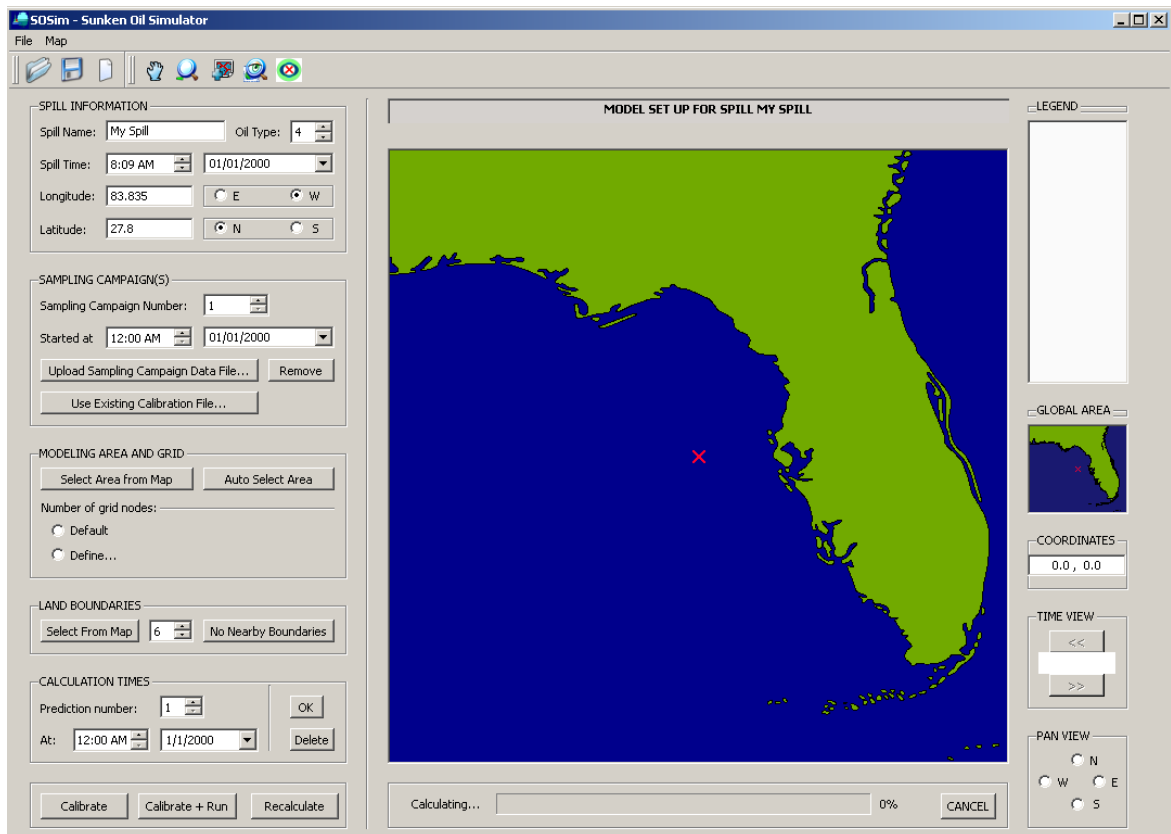


Figure 3.3. Marked spill site after selection of longitude, latitude values and directions.

3.2. Sampling Campaign(s)

Oil movement on the bottom is inferred by the model based principally on the relationship between the location and time of the spill, and the locations and relative concentrations at subsequent sampling times. For purposes of SOSim input, a sampling

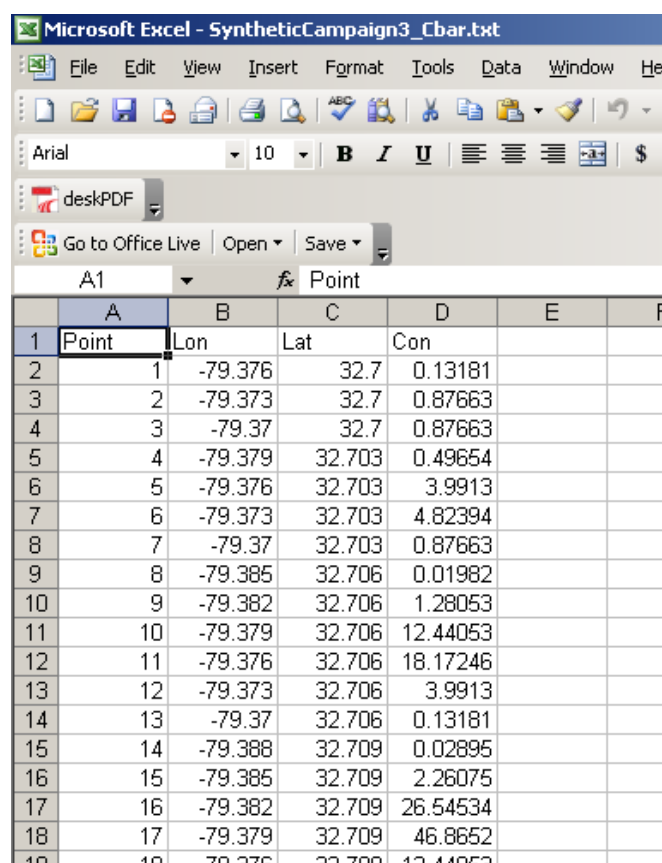
campaign is defined as a set of qualitative measurements of relative oil concentration on the bottom, taken at approximately the same time. Spill and sample coordinates are entered in WGS (World Geodesic System) units (decimal degrees of longitude and latitude). Time differences from sampling point to point of up to a few hours may not need to be considered, as the model was not designed to track possible cyclic excursions of the oil due to the tidal cycle. SOSim version 1.0rc1 accepts and has capability for processing up to 10 different sampling campaigns.

Each sampling campaign file must be created in a separate Microsoft Excel spreadsheet, closely following the instructions below:

- The spreadsheet must use only the first 4 columns: A, B, C and D
- A title (field name) for the column must be included in the first line of each column,
- In Column A, enter index numbers for the data points, beginning with the numeral “1” to be entered on line 2;
- In Column B, enter the longitude coordinate, in decimal degrees, at which the sample was collected, starting on line 2;
- In Column C, enter the latitude coordinate, in decimal degrees, at which the sample was collected, starting on line 2;
- In Column D, enter a positive or zero value for the measured relative oil concentration on a relative scale of range scale of 0 – 100 (that is, enter a

percentage of oil without the percent sign), starting on line 2. If quantitative measurements are available, measurements can be entered with accuracy of up to several decimal places.

When complete, the Excel file must be saved as a “.txt (tab-delimited)” file in the location of your preference. Figure 3.4 shows an example of the Excel file.



	A	B	C	D	E	F
1	Point	Lon	Lat	Con		
2	1	-79.376	32.7	0.13181		
3	2	-79.373	32.7	0.87663		
4	3	-79.37	32.7	0.87663		
5	4	-79.379	32.703	0.49654		
6	5	-79.376	32.703	3.9913		
7	6	-79.373	32.703	4.82394		
8	7	-79.37	32.703	0.87663		
9	8	-79.385	32.706	0.01982		
10	9	-79.382	32.706	1.28053		
11	10	-79.379	32.706	12.44053		
12	11	-79.376	32.706	18.17246		
13	12	-79.373	32.706	3.9913		
14	13	-79.37	32.706	0.13181		
15	14	-79.388	32.709	0.02895		
16	15	-79.385	32.709	2.26075		
17	16	-79.382	32.709	26.54534		
18	17	-79.379	32.709	46.8652		
19	18	-79.376	32.709	12.44053		

Figure 3.4. Creation of a sampling campaign file. Recording spatial coordinates and observed relative concentration values (scale 0 to 100) for each sampling point in an Excel file.

To upload each sampling campaign file, make use of the prompts and buttons in the Sampling Campaign(s) panel of the user interface shown in Figure 3.5.

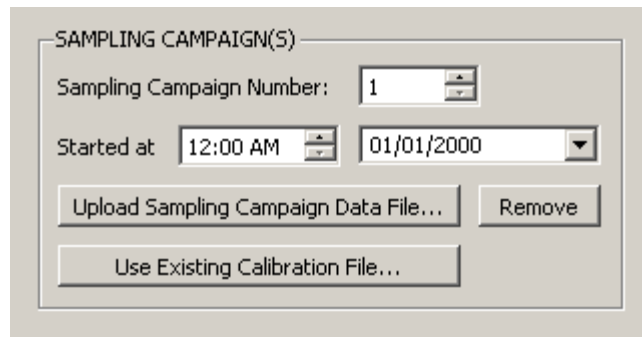


Figure 3.5. Sampling Campaign input prompts and buttons in the GUI.

The following descriptions and conditions apply:

- *Sampling Campaign Number*: adjust the number in the spin box to that of the sampling campaign that you want to upload and process. A campaign number greater than 10 is not accepted;
- *Start at: Time, Date*: set the spin box to the time representing the midpoint of the sampling campaign. Note that you can edit the hour and minutes by using either the mouse cursor or the arrows of the spin box, or both. Set the day, month, and year of the sampling campaign using the smart calendar that pops up when you click in the drop-down menu located to the right. Notice that the date line edit can also be changed manually with the mouse cursor. It is not necessary that the dates of previously uploaded campaigns have earlier sampling dates than the campaign

currently being uploaded, but the dates of all sampling campaigns must be subsequent to the spill date entered and subsequent to the assessed retardation gap, during which oil may be still sinking depending on its type. A warning message will guide you in case of error. Sampling dates of different campaigns can even be the same (different response teams can take samples in different areas at the same time). Nevertheless, it is recommended that sampling be conducted at different times in order for the model to better account for changes in sunken oil movement due to e.g. tidal action, storm events, and sediment entrainment.

- *Upload Sampling Campaign Data File* button: this button opens an explorer dialog box that allows you to browse for your “.txt (tab-delimited)” Excel file containing the sampling campaign information which you want to upload. It is recommended that every sampling campaign file have a different name and be identified with a number. If the sampling campaign entered happens to be during the assessed sinking retardation time, the sampling campaign will be invalidated, and a warning issued.
- *Remove Sampling Campaign* button: this button automatically removes from the record the data of the campaign that corresponds to the current number in the *Sampling Campaign Number* spin box. A confirmation message pops up to confirm the deletion, along with the number of the sampling campaign that was removed. However, deletion of the file does not change the numbers of any other uploaded campaigns, so that there will now be a campaign number with no assigned data.

- *Use Existing Calibration File* button: this button allows the modeler to use an existing calibration file, which must have been named and saved by the user after a previous calibration. By clicking on this button, a search dialog box will open and will prompt for the directory and file location. An informational message will confirm that the calibration file has been loaded when found.

An example uploaded sampling campaign data file is shown in Figure 3.6.

Point	Lon	Lat	Con
1	-79.37600	32.70000	0.13181
2	-79.37300	32.70000	0.87663
3	-79.37000	32.70000	0.87663
4	-79.37900	32.70300	0.49654
5	-79.37600	32.70300	3.99130
6	-79.37300	32.70300	4.82394
7	-79.37000	32.70300	0.87663
8	-79.38500	32.70600	0.01982
9	-79.38200	32.70600	1.28053
10	-79.37900	32.70600	12.44053
11	-79.37600	32.70600	18.17246
12	-79.37300	32.70600	3.99130
13	-79.37000	32.70600	0.13181
14	-79.38800	32.70900	0.02895
15	-79.38500	32.70900	2.26075
16	-79.38200	32.70900	26.54534
17	-79.37900	32.70900	46.86520
18	-79.37600	32.70900	12.44053
19	-79.37300	32.70900	0.49654
20	-79.39100	32.71200	0.02895
21	-79.38800	32.71200	2.73237
22	-79.38500	32.71200	38.77601
23	-79.38200	32.71200	82.73948
24	-79.37900	32.71200	26.54534
25	-79.37600	32.71200	1.28053
26	-79.39400	32.71500	0.01982
27	-79.39100	32.71500	2.26075
28	-79.38800	32.71500	38.77601
29	-79.38500	32.71500	100.00000
30	-79.38200	32.71500	38.77601
31	-79.37900	32.71500	2.26075
32	-79.37600	32.71500	0.01982
33	-79.39400	32.71800	1.28053
34	-79.39100	32.71800	26.54534
35	-79.38800	32.71800	82.73948
36	-79.38500	32.71800	38.77601
37	-79.38200	32.71800	2.73237
38	-79.37900	32.71800	0.02895

Figure 3.6. Example uploaded sampling campaign data file.

3.3. Modeling Area and Grid

The modeling area is the geographical area over which the user wishes to predict oil locations in time. Due to limitations in terms of computational time and the corresponding resolution of the statistical computations and output, assignment of modeling areas larger than 50 km wide by 50 km long (approximately 0.50 longitude by 0.5 latitude degrees) is not recommended. However, assignment of larger regions is allowed by SOSim, for example to allow the user to first check the overall movement or direction of a spill before modeling smaller areas. In that case, a warning message will pop up requesting permission from the user to assign such a large area, and warning of the disadvantage in doing so. An alternate approach for viewing results over a larger area, is conduct a run for the principal area first, then conduct partial re-runs of the program for contiguous regions using the post processing “*Pan-posteriori*” and “*Recalculate*” set of tools located to the right of the screen, once their use become available after processing the current run. However, in this case the results for each area modeled will have a new color scale, not continuous with the first (because the highest relative concentration predicted for each run will be colored dark red, for example, though these concentrations will not likely be equal), making interpretation more difficult.

The grid is a set of orthogonal points in the west-east and north-south directions defined on a Cartesian plane, representing the area to be modeled. Results of the prediction are calculated and plotted at every node in the grid. The more nodes requested in each direction, the better the spatial resolution of the mapped output and the longer the

run time. These characteristics are entered in the panel “Modeling Area and Grid” shown in Figure 3.7:

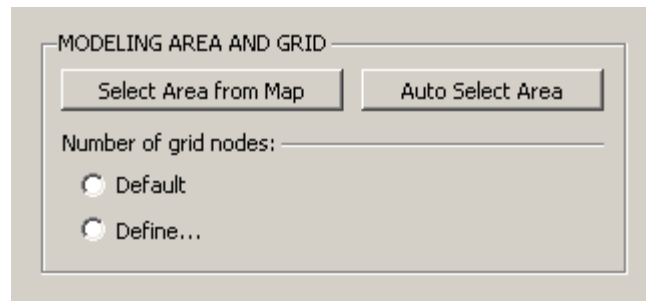


Figure 3.7. Modeling Area and Grid input prompts and buttons in the GUI.

Information is entered as follows:

- *Select Area from Map* button: this button is a map tool that enables a selection cursor. To select the modeling area, left click on the map point that corresponds to the north-east corner of the desired modeling area. Then drag the pointer, without releasing the left button, to the south-east corner of the area to be mapped. Release the mouse left button to set the area (Figure 3.8). It is not necessary to include the spill site (marked with an X) within the modeling area, though it may be helpful to model the vicinity around the spill site first;

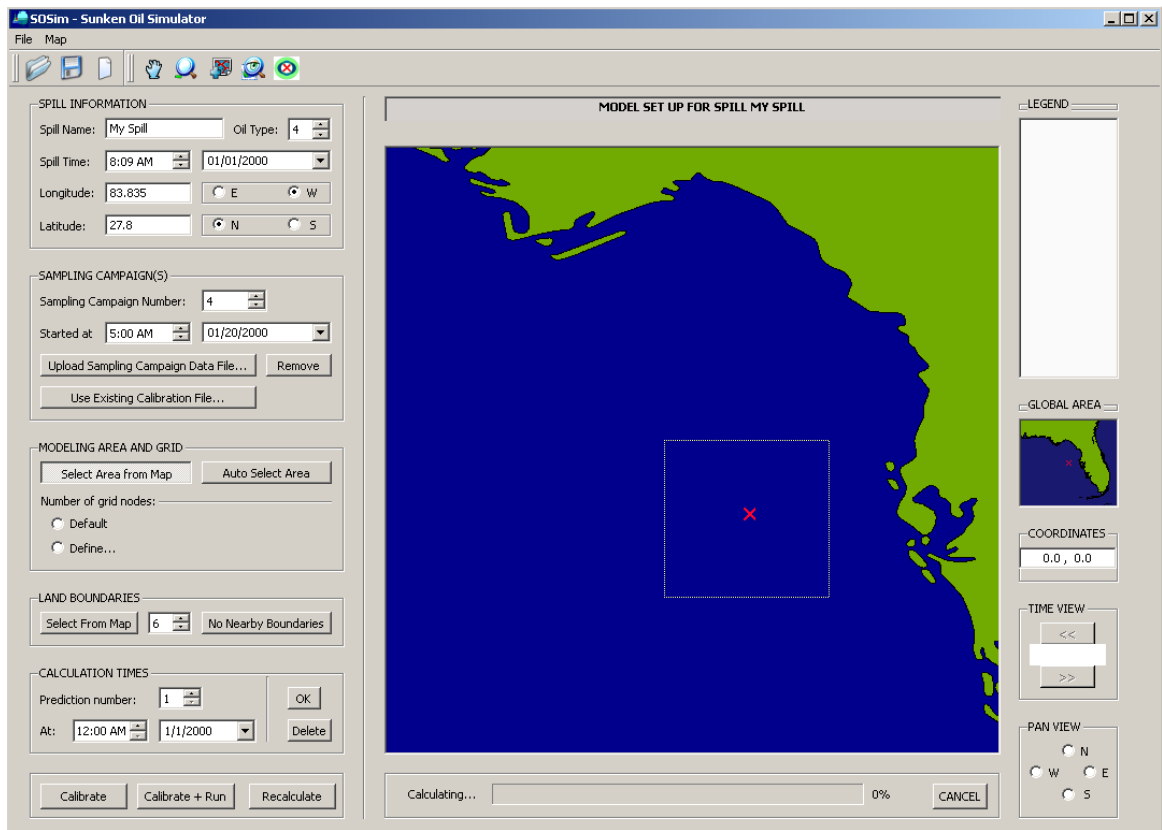


Figure 3.8. Selecting the modeling area in the GUI.

- *Auto Select Area* button: this button automatically zooms in and selects a region of 0.25 degrees longitude by 0.25 degrees latitude around the spill site (approx. 25 by 25 km or approx. 15.6 by 15.6 miles). This feature can be used along with the post processing “*Pan-posteriori*” and “*Recalculate*” set of tools when the user plans to consecutively run SOSim for contiguous geographical regions of the same size so as to obtain a mosaic of a large area; and

- *Number of Grid Nodes*: as mentioned previously, resolution of the output is defined by the number of nodes desired in the west-east and north-south directions within the modeling area. Two options are available to the user. The first is the *Default* radio button, which assigns 25 nodes in each direction. Using this option, if the modeling area is been defined using the *Auto Select Area* button, then a node is placed every 0.01 degrees (approximately every kilometer). The second is the *Define* radio button, which makes available one prompt line in each direction, for the user to assign equal or different numbers of east-west and north-south nodes.

3.4. Land Boundaries

Shoreline boundaries are accounted for by SOSim if the user indicates that land is present in the area to be modeled. Two options are available to the user, as indicated in Figure 3.9:

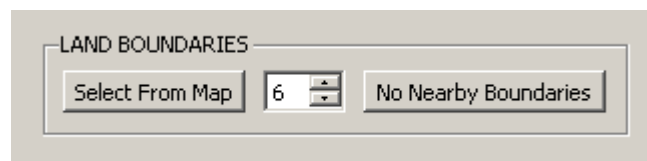


Figure 3.9. Land Boundaries input buttons and spin box in the GUI.

Limitations on the boundary conditions accepted by SOSim v. 1.0rc1 include (a) only one mass of land can be defined in each modeling area (e.g. a single polyline within a

single modeling area), (b) irregularly-shaped geographic features close to shore, such as islets and mangroves, should be considered part of a single land mass, and (c) the maximum number of vertices is set at 10. This latter restriction is not inherent to internal model computations, but was fixed so as to limit run times on desktop computers to the order of one day. Input is entered as follows:

- Boundaries spin box: by default, this box is set to 6, but can spin from 2 to 10. The number set in this box is the number of vertices that the user wishes to select on the map so as to mimic the boundary conditions by the use of a polyline. If the number is not changed before clicking on the *Select from Map* button, the number of vertices defaults to 6.
- *Select from Map* button: clicking on this button enables two functionalities. First is a message box that instructs the user to "Please select X points on the land border shown on the canvas." where X is the number in the spin box, 6 by default. Straight shorelines can be approximated by two points, slightly curved shorelines may be approximated with three points, and so on, offering the potential to greatly reduce run time. Second, the map tool with which the user defines these points is enabled. To select each point, click the left mouse button at a point on the shoreline depicted in the GUI canvas, release it in the same position, and then proceed to the next point. A thin yellow line will follow and connect these points to create a 2 to 10-vertex polyline approximating the actual boundary. Note that if a very large modeling area is selected that includes land, but all shorelines are far

from the spill and dispersing oil, the predicted results will be negligibly different from the results that could be obtained much more quickly by selecting the “*No Nearby Boundaries*” option; and

- *No Nearby Boundaries* button: this button communicates to the model the lack of need for computing reflection of sunken oil from shoreline boundaries within the modeling area, for example when the geographic area to be modeled includes only open sea.

3.5. Prediction Times

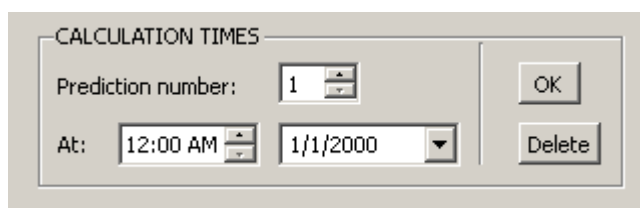


Figure 3.10. Prediction Times input buttons in the GUI.

The selection of prediction times is the last segment of required input. In order to enter these times, all previous input sections must be completed. An error message pops up if input information is missing. Projection times are the dates and hours the user wishes to view predicted sunken oil relative concentrations. The user can request up to five times of prediction per model run. However, computational run time will be longer for each projection time requested, and it is recommended that one time at a time is

modeled with the aim to plan and make timely decisions about subsequent run needs. Times are input to the panel shown in Figure 3.10.

SOSim v. 1.0rc1 calculations are based on the assumption that sunken oil impacting the shoreline will be reflected back into the water. This assumption allows modeling of accumulation at the shoreline. However, the assumption may not be realistic at times after which the majority of the oil mass is predicted to reach the nearshore environment. At these later times, predictions may indicate a “bouncing” of the patch off of the shoreline. To attempt to avoid this potentially unrealistic result, a warning message will pop up if the requested prediction time is estimated by the model *a priori* to be after the time of predicted landfall. The user should be aware that such predictions in the nearshore environment may potentially occur, even in some cases if no *a priori* warning is issued. In general, predictions of substantial sunken oil occurrence at the shoreline should be interpreted with care.

Input is entered as follows:

- *Prediction number* spin box: Assign a whole number index to each desired time of prediction. If the index number is not changed in the spin box when adding a new date and hour, a warning message will appear;
- *At* hour spin box and time line prompt: Complete these fields with, respectively: (a) the hour, including minutes if desired, and (b) the date, from the smart

calendar pop-up, at which predictions are desired. The user must confirm final entry by clicking the *OK* button;

- *OK* button: Add the current hour and time to a list of prediction times, and assign it to the index position indicated by the “*Prediction number*” spin box. The GUI will then estimate whether the time entered is likely to be within the time required for the oil mass to reach the shoreline, if the oil is apparently moving towards the shore. If the time entered is determined to be potentially after the time of landfall, a warning message will pop up. Any prediction time prior to the most current sampling campaign cannot be modeled as the likelihood is affected by all sampling campaigns. A message will appear requesting that only sampling campaigns prior to the requested prediction time be on file.
- *Delete* button: If desired, you can use this button to cancel the addition of the prediction time that corresponds to the current value in the “*Prediction number*” spin box. If a prediction time is deleted by mistake, the hour and date can be added using a different, new index or “*Prediction number*”.

3.6. Default Input

Default input corresponds to internal parameter ranges that may be modified by an expert in oceanography or Bayesian modeling with the aim to optimize the precision of the predictive capabilities of the simulator based on the changing conditions of the modeling scenario or the specifics of a known situation given the occurrence of an oil

spill. Default input may be modified every time that SOSim is launched if the default ranges need to be changed.

To modify the default input, go to the *Options* menu of SOSim and click on *Modify Default Settings*. A tabbed widget will guide you through the 3 settings that can be modified, including the default initial parameters range, the default number of grid nodes and the default resolution. Change the settings as required; observing that some values are restricted by the software given mathematic and physical constraints. Figure 3.11 shows the preview of the *Parameters Range* tab.

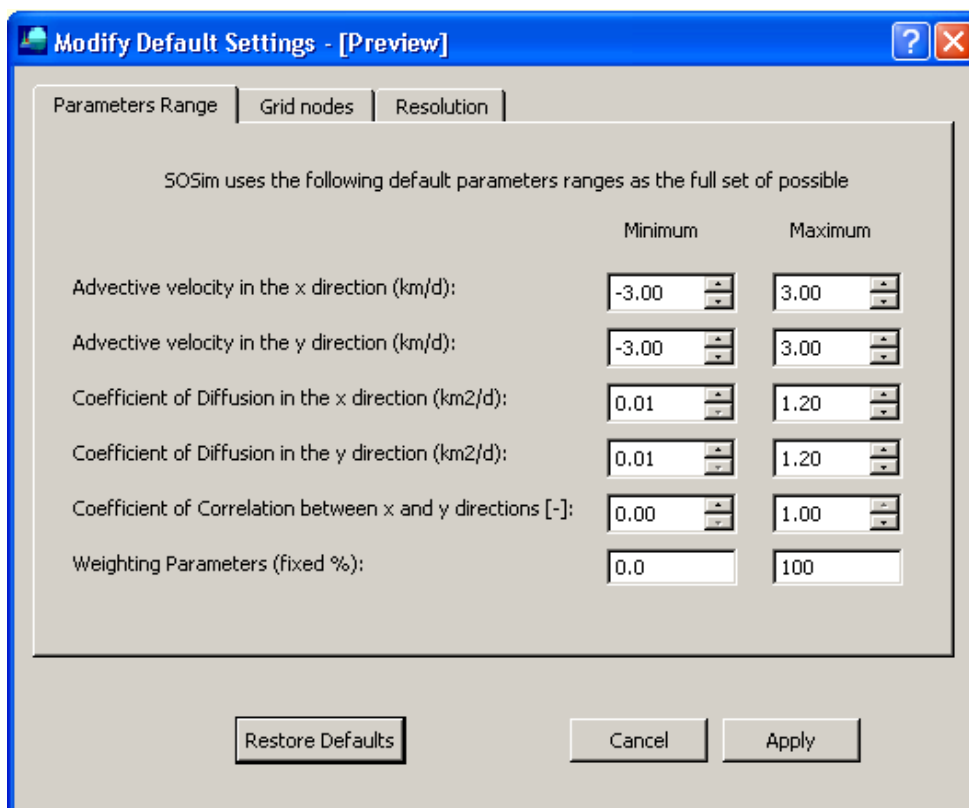


Figure 3.11. Modify default input settings dialog box.

4. Processing

This section describes operation of the model from the moment that the input information is complete until production of the default output requested by the user. Output can be (1) only a calibration file to be saved for further processing, (2) a calibration file plus map predictions for requested scenarios based on the last, or (3) recalculations based on previous predictions and a saved or recently created calibration file. Figure 4.1 shows the processing buttons of the GUI.

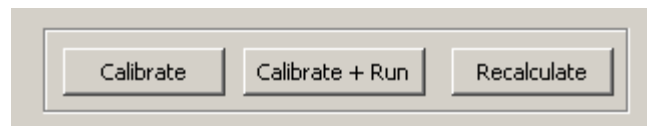


Figure 4.12. Processing buttons

4.1. The Calibrate, Calibrate + Run, and Recalculate Buttons

Calibrate button: this button is used if it is intended only to calibrate the model by using only the sampling campaign(s) uploaded files, the spill information, and the characteristics of the geographic area. No prediction times need be selected to run a calibration. When finished, the calibration files can be saved by the user for future use.

Calibrate + Run button: this button is used to instruct SOSim to first calibrate the model using the sampling campaign files and then immediately begin computations for prediction. This button is intended to obtain the results of a complete scenario by investing time in only one initial configuration. Overnight calibration-and-runs sessions are possible using this button. No changes to the input are allowed at this point until the core module has completed the Bayesian processing stage.

Recalculate button: this button has several uses. First, the button may be used to generate results for the same spill scenario at a different time prediction, without data entry and recalibration. Second, the button may be used to obtain a new prediction following a change in resolution, boundaries, or geographical area (post processing and optional output), again without additional data entry or recalibration. Third, the button can be used to generate results using an existing calibration file that has been imported using the “*Use Existing Campaign*” button, in the Sampling Campaign(s) section.

4.2. Run Time and Progress Bar

Run time increases as function of (a) the number of output nodes, or geographical resolution, selected by the user, (b) the existence of boundary conditions within the selected modeling area, and (c) the number of vertices in the polyline representing the shoreline, if present. A longer time of calculation (about 5 times longer) should be allowed when modeling spills within a coastal area. The progress bar indicates the total number of operations completed.

5. Output

Output is produced as a map representation of predicted relative oil concentrations within the modeling area. There are two classes of output: the default and the optional output.

5.1. Default Output

The map at the first requested time of prediction is automatically displayed in the GUI canvas immediately after the processing stage is completed. In addition, a portable raster map (PNG format) of this result is saved to the desktop, along with a corresponding world file required for automatic georeferencing in any geographic information system. If more than one time of prediction was initially requested, map results corresponding to all prediction times will be available for the user to display on the canvas and save to disk, as the user requests with the “*next*” and “*previous*” buttons located to the right of the display.

Output maps are color-coded such that dark red areas are predicted to have high relative sunken oil concentrations (near 100), and dark blue areas are predicted to have low relative concentrations (near zero). Predicted relative concentrations are effectively averaged across smaller scale patchiness of the sunken oil, and across cyclic movement due to tidal excursions. Also, total sunken oil masses are not known as a function of time.

In fact, although relative concentrations may decrease with time due to oil dispersion, total sunken oil mass may simultaneously be changing due to sinking and re-suspension. Therefore, colors are not related from one time of prediction to another (that is, the same colors in different maps do not indicate the same relative concentration values). Rather, a scale of dark red to dark blue is presented at each time, indicating relative spatial probabilities of finding sunken oil at each time independently. The same interpretation applies whether individual prediction times are requested as part of the same run or in separate runs. It should be noted that the internal calculations of SOSim assume a constant total sunken oil mass in time, so as to maintain internal consistency and quality control checks on the integration computations. However, this assumption does not affect model output.

5.2. Optional Output

Optional output includes (1) map results for areas that are either contiguous to the current modeling area, zoomed-in within the current modeling area, or zoomed-out to include a larger area; and (2) maps updated with new or revised sampling campaign data. Optional output is obtained as follows.

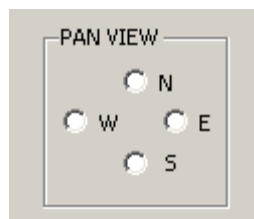


Figure 5.13. Pan View button set

To model areas contiguous to a current output map:

- Under the section “*Pan View*” (Figure 5.1), select the direction in which the desired contiguous region is located with respect to the current output map in the display. For example, if the desired new modeling area is the continuation of a coastal region that extends to the north, select the button “N” under the “*Pan View*” section.

The canvas will move towards an area of the same map scale and size in the selected direction;

- A pop-up message will request the user to complete the boundary conditions for the new area, along with the desired resolution (number of nodes in the east-west and south-north directions) for this contiguous case. Of course, any change in resolution may result in changes to the grid of the output map. If the new area is being modeled with the intention of creating a mosaic of multiple runs, a constant resolution is recommended;
- Prediction times can be changed, but be aware that in order to see the continuation of the original output, the same times of prediction are required; and
- Press the “*Recalculate*” button in the “*Pan View*” panel. Pressing this button instead of the “*Run*” button will instruct SOSim to use previous input information

regarding the sampling campaigns and the spill, and will require much shorter runtime than would be required to rerun the case from scratch using the “*Run*” button.

To model zoomed-in areas of a current output map, or larger zoomed-out areas:

- From the panel “*Modeling Area*” press the “*Select Modeling Area*” button. Use the tool as before, making sure to select the resolution (number of nodes in each direction) and new boundary conditions desired before starting the run. The “*Zoom in*” tool under the “*Map*” toolbar also works to select a modeling area, if the user subsequently confirms the nodes and boundaries;
- Change times of prediction, if desired; and
- Press the “*Recalculate*” button under the “*Pan View*” section. Pressing this button instead of the “*Run*” button will instruct SOSim to use previous input information regarding the sampling campaigns and the spill, and will require much shorter runtime than would be required to rerun the case from scratch using the “*Run*” button.

To update the current output map based on new or revised sampling campaign data:

- If a sampling campaign is no longer needed or requires revision, remove it using the “*Remove Sampling Campaign*” button, making sure to select the correct

campaign number from the drop-down menu; upload the new sampling campaign data as usual; and

- If the modeling area, boundaries, and prediction times are not to be changed, then press the “Run” button. The “Recalculate” button is not useful in this case because the model run must be computed from scratch using the new field data, conserving only the desired prediction date(s), areas, boundaries and initial spill information.

6. Post Processing

Post processing activities include uploading previous raster output for viewing in a GIS (geographic information system), saving output, saving calibration files for further use, and printing images. The post processing tools are located in the “*Map*” toolbar of the GUI. SOSim Version 1.0rc1 provides only for saving the output of a current project, but not for saving the contents of the prompts to be opened as a project later.

7. Portability of Results

To allow the posting of SOSim output data to the World Wide Web for rapid, long-distance, widely-accessible transfer of information, output image data is produced and saved in Portable Network Graphics (PNG extension) format. PNG is an extensible file format for the lossless, portable, well-compressed storage of raster images. As a complement of any output map created by SOSim, a corresponding world or georeferencing file (with PGW extension) is created and saved. The PGW file must accompany its parent PNG raster file in order for any output map to be ported to any GIS (geographic information system) that supports common PNG raster images, including but not limited to ArcGIS, QGIS, GRASS, and SOSim itself. PNG figures can be imported, inserted, printed, and used like any other common image file.

8. Software Portability

SOSim version 1.0rc1 comes in an executable package for the Microsoft Windows 32 bit platform, allowing easy portability among computers having this common operating system. Portability to other OS platforms may be considered for future versions.

REFERENCES

- Echavarria Gregory, M. Angelica. 2010. Predictive Data-Derived Bayesian Statistic-Transport Model and Simulator of Sunken Oil Mass. Doctoral Dissertation. University of Miami, July 2010.
- Englehardt, James D., Echavarria Gregory, M. Angelica; Avellaneda, Pedro. 2010. Development of a Predictive Bayesian Data-Derived Multi-Modal Gaussian Maximum Likelihood Model of Sunken Oil Mass. Draft Final Report, June 2010.
- Research Planning Inc. 1994. Natural Resource Damage Assessment Emergency Guidance Manual. Version 1.1, Columbia, SC, 1994.