

# Universidad Nacional Autónoma de México

## Centro de Ciencias de la Atmósfera

### Software: *Oil spill v. struct*

### Guía de usuario

#### Desarrollado por:

Grupo interacción océano-atmósfera [<http://grupo-ioa.atmosfera.unam.mx/>]

Jorge Zavala Hidalgo [[jzavala@atmosfera.unam.mx](mailto:jzavala@atmosfera.unam.mx)]

Julio Antonio Lara Hernández [[brn2fight@live.com](mailto:brn2fight@live.com)]

Olmo Zavala Romero [[olmozavala@atmosfera.unam.mx](mailto:olmozavala@atmosfera.unam.mx)]

Fernando Nicolás Arellano Guerrero [[fnag@atmosfera.unam.mx](mailto:fnag@atmosfera.unam.mx)]

Rosario Romero Centeno [[rosario@atmofera.unam.mx](mailto:rosario@atmofera.unam.mx)]

Ma Elena Osorio Tai [[tai@atmosfera.unam.mx](mailto:tai@atmosfera.unam.mx)]

Septiembre, 2017

# Índice de contenido

<b>1. Introducción.....</b>	<b>3</b>
<b>2. Requerimientos.....</b>	<b>3</b>
2.1. Matlab.....	3
2.2. Campos de velocidad.....	3
<b>3. Contenido del software.....</b>	<b>4</b>
<b>4. Ejecución del software.....</b>	<b>4</b>
4.1. Modo gráfico.....	4
4.2. Modo programático.....	4
4.2.1. Caso BP, 2010.....	4
4.2.2. Caso Usumacinta, 2007.....	5
4.2.3. Otros casos.....	6
4.2.3.1. Edición del archivo local_paths_X.m.....	6
4.2.3.2. Edición del archivo CSV.....	6
4.2.3.3. Edición del archivo main_X.m.....	6
4.2.3.4. Edición de la rutina velocityFields_N.m.....	7
<b>5. Resultados.....</b>	<b>7</b>
<b>6. Referencias.....</b>	<b>9</b>
<b>Apéndice A: algoritmo general del modelo (oilSpillModel.m).....</b>	<b>10</b>

## 1. Introducción

El objetivo del software *Oil spill v. struct* es simular derrames de petróleo en el Golfo de México, tales como el derrame BP, 2010 “Deepwater Horizon” o el derrame de la plataforma Usumacinta, 2007. El software está programado modularmente con estructuras de Matlab para facilitar su comprensión e implementación.

Para simular un derrame de petróleo se usa un algoritmo Lagrangiano de rastreo de partículas que requiere campos archivados de velocidad de las corrientes oceánicas y del viento en superficie. Al viento se le añade un ángulo de deflexión para considerar el efecto Coriolis (Samuels *et al.*, 1982), y a las velocidades se les añade un componente aleatorio para considerar el efecto de difusión-turbulenta (Döös *et al.*, 2011). Las simulaciones se pueden realizar a distintas profundidades, considerando uno o varios componentes de petróleo (Le Hénaff *et al.*, 2012), distintos tipos de decaimiento (natural, químico, evaporación, quema, recolección, y/o degradación exponencial) (Oil Budget Calculator Science and Engineering Team, 2010), el paso de tiempo deseado, y un método de integración Runge-Kutta de 2° o 4° orden. Para la liberación y eliminación de partículas se usa un enfoque probabilístico que previene errores por redondeo. El modelo además puede generar la visualización de las simulaciones en dos y tres dimensiones.

Esta guía detalla el contenido del software *Oil spill v. struct*, y proporciona la información necesaria para realizar distintas simulaciones de derrames de petróleo en el Golfo de México.

## 2. Requerimientos

### 2.1. Matlab

La totalidad de *Oil spill v. struct* está escrita en Matlab, por lo que puede ser ejecutado en cualquier sistema operativo con una licencia de Matlab vigente. El software se probó en las versiones de Matlab 2014b – 2017a para Windows y Linux.

### 2.2. Campos de velocidad

Es necesario contar con archivos de campos de velocidad de las corrientes y del viento. Estos datos pueden provenir de modelos de circulación oceánica como el HYCOM (Hybrid Coordinate Ocean Model) y de modelos atmosféricos como el WRF (Weather Research Forecast). Datos de corrientes oceánicas generados con el modelo HYCOM se pueden descargar gratuitamente de <http://hycom.org>.

### 3. Contenido del software

*Oil spill v. struct* es un conjunto de códigos y archivos distribuidos en distintas carpetas. La carpeta principal lleva por nombre “OilSpill\_vStruct” y su contenido es el siguiente:

#### - OilSpill\_vStruct

**graphical\_interface.fig** -> Archivo binario con la figura de la interfáz gráfica.

**graphical\_interface.m** -> Ejecuta el modelo en modo gráfico.

**main\_BP\_50p1.m** -> Ejecuta la simulación del derrame BP en modo programático.

**main\_Usumacinta.m** -> Ejecuta la simulación del derrame Usumacinta en modo programático.

**local\_paths\_BP\_50p1.m** -> Indica los directorios de entrada para la simulación de BP.

**local\_paths\_Usumacinta.m** -> Indica los directorios de entrada para la simulación de Usumacinta.

+ **docs** -> Contiene una presentación y la guía de usuario.

+ **data** -> Contiene las batimetrías (.mat y .nc) y un archivo CSV con datos del derrame BP.

- **lib** -> Contiene las rutinas (.m) para ejecutar al modelo.

+ **external** -> Rutinas ajenas para guardar figuras (export\_fig).

+ **model** -> Rutinas principales del modelo.

+ **plotting** -> Rutinas para leer batimetrías (get\_bathymetry) y graficar (map\_particles, plotStats).

+ **tools** -> Rutinas auxiliares utilizadas por las rutinas principales.

### 4. Ejecución del software

#### 4.1. Modo gráfico

Para iniciar el modelo en modo gráfico, la rutina *graphical\_interface.m* se tiene que ejecutar con Matlab. La interfáz gráfica (Fig. 1) tiene una configuración predeterminada para simular el derrame BP, 2010; no obstante, esta se puede adaptar para simular algún otro derrame.

#### 4.2. Modo programático

##### 4.2.1. Caso BP, 2010

La rutina *main\_BP\_50p1.m* está configurada para simular el derrame de petróleo BP, 2010 “Deepwater Horizon” a las profundidades de 0, 400, y 1100 m. Para realizar esta simulación primero se necesita indicar la ruta de los archivos de entrada en la rutina *local\_paths\_BP\_50p1.m*. Posteriormente hay que iniciar el programa Matlab, abrir el script *main\_BP\_50p1.m*, indicar el directorio donde se deseen guardar los resultados, y finalmente presionar el botón de ejecutar. Adicionalmente, se pueden cambiar los parámetros en el script *main\_BP\_50p1.m* para personalizar la simulación.

**graphical\_interface**

**Output directory**  
/home/user/Desktop/OilSpill\_vStruct/Results/BP/

**Input directories file**  
local\_paths\_BP\_50p1.m

**Initial spill day**  
2010,04,22

**Final spill day**  
2010,07,14

**Final simulation day**  
2010,07,30

**Spill location [lat, lon]**  
28.738, -88.366

**Domain limits**  
xmin, xmax, ymin, ymax, zmin, zmax  
-92, -80, 25, 31, -2500, 0

**Bathymetry File**  
BAT\_FUS\_GLOBAL\_PIXEDIT\_

**Time steps in hours**  
Lagrangian, Ocean, Wind, Saving  
1, 24, 6, 24

**Runge-Kutta, V. fields type**  
4, 1

**Oil classes (fractions)**  
0.05, 0.20, 0.30, 0.20, 0.10, 0.10

**Simulation depths:**  
1100, 400, 0

**Initial spill radius (m) per depth:**  
250, 500, 1000

**Turbulent-diffusion per depth:**  
0.2, 0.8, 1.0

**Wind contribution (for 0 m depth):**  
0.035

**Oil decay**

☒ Evaporation  
☐ Natural  
☐ Chemical  
☐ Collection  
☐ Burning --> Burning radius (m): 300000

☒ Exponential degradation:  
95 %  
3, 6, 9, 12, 15, 18, 21, 24 Days per oil class

**Daily quantities**

Particles per barrel: 1

CSV File (filename.csv | 0)  
spill\_data.csv

Spilled barrels: 58668

Collected barrels: 7713

Burned barrels: 2655

Chem. dispersants (gallons)  
Surface, Subsurface  
50, 50

Developed by: Grupo Interacción Océano-Atmósfera, CCA, UNAM.  
<http://grupo-ioa.atmosfera.unam.mx/>

**Start**

**Figura 1.** Modo gráfico del modelo Oil Spill v.Struct.

#### 4.2.2. Caso Usumacinta, 2007

La rutina *main\_Usumacinta.m* está configurada para simular el derrame de Usumacinta a las profundidades de 0, 5, y 10 m. Para realizar esta simulación primero se necesita indicar la ruta de los archivos de entrada en la rutina *local\_paths\_Usumacinta.m*. Posteriormente hay que iniciar el programa Matlab, abrir el script *main\_Usumacinta.m*, indicar el directorio donde se deseen guardar los resultados, y finalmente presionar el botón de ejecutar. Adicionalmente, se pueden cambiar los parámetros en el script *main\_Usumacinta.m* para personalizar la simulación.

### 4.2.3. Otros casos

Para simular otros eventos de derrame de petróleo, el usuario tendrá que editar los archivos CSV, `velocityFields_N.m`, y/o `main_X.m`, según las características del derrame que desee simular y los campos de velocidad a utilizar. Por último, solo se necesitará ejecutar el archivo `main_X.m`. Se recomienda conservar los archivos originales.

#### 4.2.3.1. Edición del archivo `local_paths_X.m`

Es necesario crear un archivo `local_paths_X.m` en donde se especifique la dirección de los archivos de entrada; es decir, la ubicación de los campos de velocidad de corrientes oceánicas y de viento que se deseen utilizar.

#### 4.2.3.2. Edición del archivo CSV

En la rutina `main_X.m` el usuario puede indicar si se ocupará o no algún archivo CSV. El archivo CSV se ubica en la carpeta *data*, y es útil en caso de que se desean derramar y/o degradar cantidades de petróleo que varíen día a día. Si se desean derramar y degradar cantidades constantes (e.g. promedios diarios), estas se indican en el archivo `main_X.m`, y no hace falta ocupar un archivo CSV. En el archivo CSV se especifican los barriles de petróleo (bbl) diariamente descargados, quemados, recolectados directamente del pozo (RITT/TopHat), y recolectados del agua (oily water), así como el número de galones (gal) de dispersantes químicos inyectados en la superficie y subsuperficie.

Ejemplo de archivo CSV:

Month	Day	Year	Discharge (bbl)	Inland Recovery (T)	Burned (bbl)	RITT_TopHat (bbl)	OilyWater (bbl)	Subsurf Dispersants (gal)	Surface Dispersants (gal)
4	20	2010	0	0	0	0	0	0	0
4	21	2010	0	0	0	0	0	0	0
4	22	2010	62200	0	0	0	0	0	1701
4	23	2010	62100	0	0	0	1630	0	0
4	24	2010	61900	0	0	0	155	0	0

#### 4.2.3.3. Edición del archivo `main_X.m`

Aquí el usuario define las características principales de la simulación a realizar:

- Tiempo y ubicación del derrame.
- Carpeta donde se guardarán los resultados.

- Nombre de la rutina (local\_paths\_X.m) que indica la ubicación de los archivos de entrada (e.g. Campos de velocidad).
- Método Runge-Kutta (de 2° o 4° orden) y tipo de campos de velocidad a utilizar.
- Límites del dominio.
- Número de partículas asignadas a cada barril.
- Grado de difusión-turbulenta por profundidad.
- Contribución del viento en superficie.
- Fracción del petróleo en subsuperficie asignada a cada profundidad de subsuperficie.
- Número de componentes de petróleo y su abundancia relativa.
- Resolución temporal de los campos de velocidad.
- Paso de tiempo Lagrangiano.
- Decaimientos a considerar (quema, recolección, natural, químico, y/o degradación exponencial).
- Si se utilizará algún archivo CSV, o no.
- Características de la generación de mapas y estadísticas.
- Especificaciones para el guardado de datos.

#### 4.3.3.4. Edición de la rutina velocityFields\_N.m

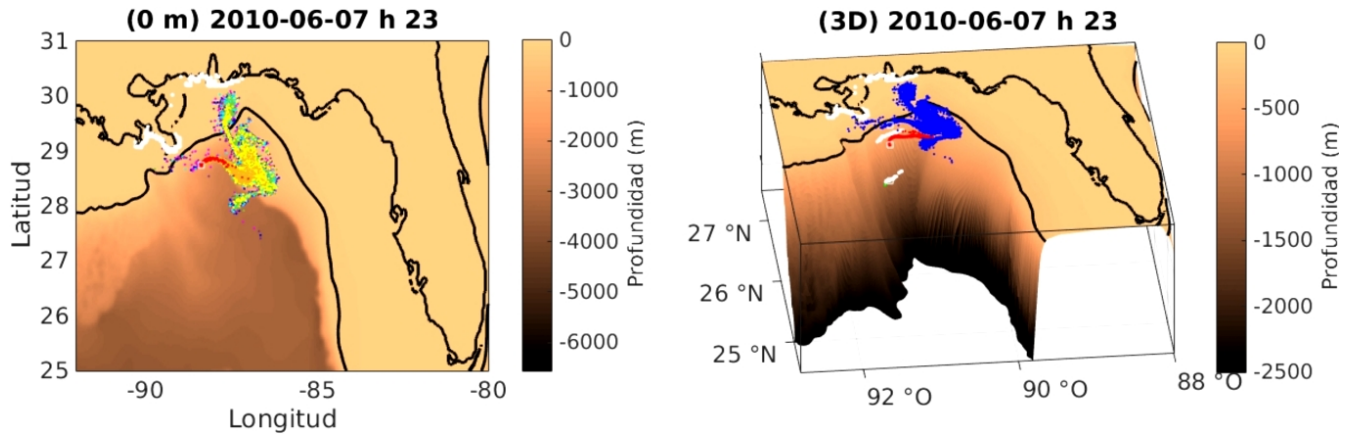
Esta rutina lee los campos de velocidad a utilizar. Aquí se especifica el nombre de los archivos y de las variables que contienen (e.g. Latitud, longitud, u, v, ...), así como la forma correcta de leerlos en función de su resolución espacio-temporal. El usuario debe adaptar esta rutina conforme a las características de sus datos de velocidad.

## 5. Resultados

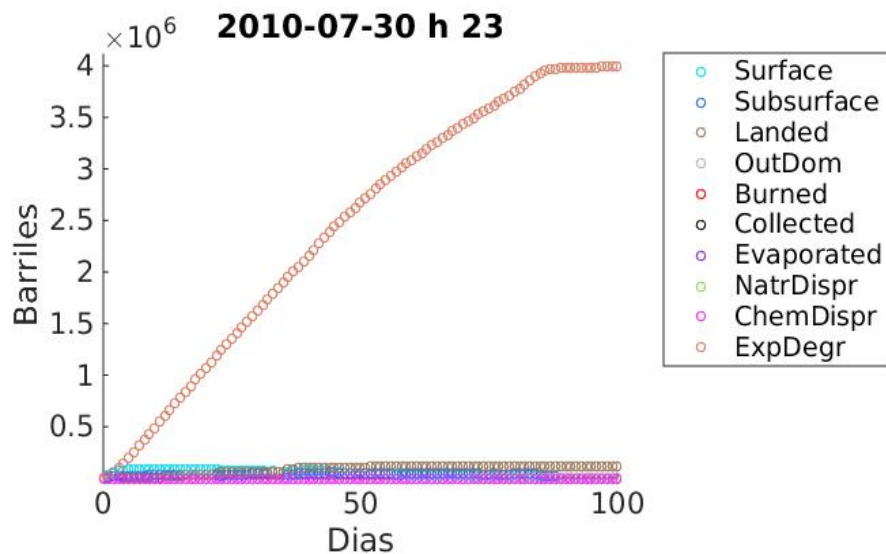
Durante las simulaciones se pueden ir presentando mapas en vista 2D y/o 3D con la posición de las partículas a determinado tiempo (Fig. 2). El color de las partículas indica el componente de petróleo que representan, y estas se marcan de color blanco cuando llegan al continente. Durante las simulaciones también es posible presentar un gráfico de estadísticas indicando las distintas cantidades de petróleo tras cierto tiempo (Fig. 3). Además, al final de la simulación se pueden generar mapas con las zonas oceánicas más frecuentadas por cada componente de petróleo (Fig. 4), así como un mapa indicando el

número total de barriles de petróleo que llegaron a la línea de costa (Fig. 5). Todas las figuras se pueden guardar como imágenes individuales y/o como videos según se especifique.

Si se seleccionó la opción de guardar los resultados numéricos de las partículas (latitud, longitud, etc.), dentro de la carpeta de resultados se generará la subcarpeta *Data*. Esta subcarpeta contendrá los archivos de Matlab (.mat) con la información de las partículas al paso de tiempo indicado.

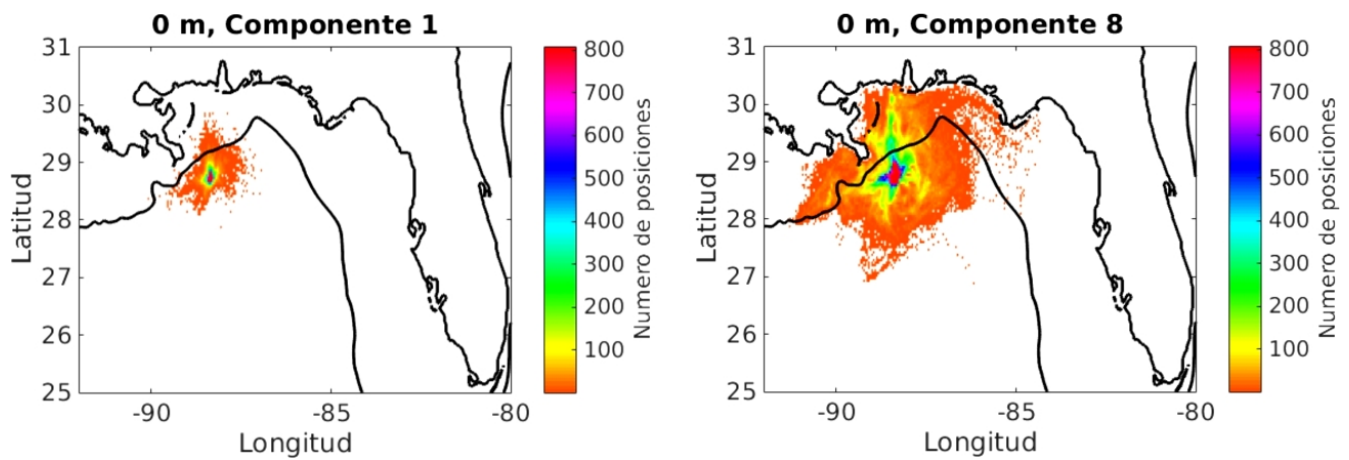


**Figura 2.** Mapas en vista 2D (izquierda) y 3D (derecha) mostrando la posición de las partículas de petróleo a las 23 horas del 7 de Junio de 2010. En la vista 2D el color indica el componente de petróleo, mientras que en la vista 3D el color indica la profundidad de las partículas.

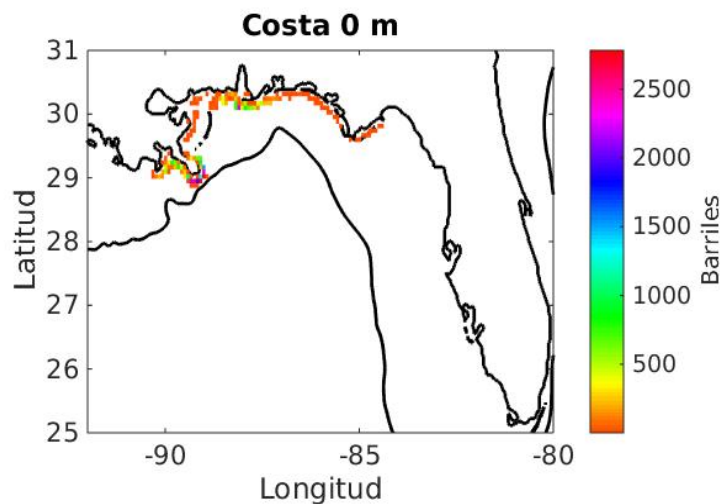


**Figura 3.** Estadísticas mostrando los barriles de petróleo en superficie, subsuperficie, tierra, afuera del dominio, quemados, recolectados, evaporados, dispersados natural y químicamente, y degradados exponencialmente. La imagen corresponde a una simulación con degradación exponencial como único decaimiento.





**Figura 4.** Mapas de las regiones oceánicas más frecuentadas por los componentes 1 y 8 del petróleo.



**Figura 5.** Mapa indicando el número de barriles que llegaron a distintas partes de la línea de costa.

## 6. Referencias

- Döös K, Rupolo V, Brodeau L (2011) Dispersion of surface drifters and model-simulated trajectories. *Ocean Model* 39: 301-310.
- Le Hénaff M, Kourafalou VH, Paris CB, Helgers J, Aman ZM, Hogan PJ, Srinivasan A (2012) Surface evolution of the Deepwater Horizon oil spill patch: combined effects of circulation and wind-induced drift. *Environ Sci Technol* 46: 7267-7273.
- Oil Budget Calculator Science and Engineering Team (2010). Oil budget calculator: Deepwater Horizon. Technical documentation. Washington, D.C.: Federal Interagency Solutions Group.
- Samuels WB, Huang NE, Amsiuz DE. (1982) An oilspill trajectory analysis model with a variable wind deflection angle. *Ocean Engng* 9: 347-360.

## Apéndice A: algoritmo general del modelo (oilSpillModel.m)

Crea las carpetas donde se guardarán los resultados.

Verifica el paso de tiempo Lagrangiano (e.g. Que sea divisor de 24 h) y lo ajusta si es necesario.

Guarda el archivo de texto “main\_config.txt” con la configuración empleada para la simulación.

Calcula constantes e inicializa variables.

**Ciclo 1.** Para cada día de simulación.

**Ciclo 2.** Para cada paso de tiempo.

**Condición 1.** Si el día actual es menor o igual al último día de derrame.

Libera nuevas partículas para cada componente del petróleo.

**Termina Condición 1.**

**Condición 2.** Si se desean visualizar y/o guardar mapas.

Genera mapas con las posiciones de las partículas de petróleo.

**Termina Condición 2.**

**Condición 3.** Verifica el tipo de los campos de velocidad a utilizar.

Obtén los campos de velocidad a utilizar.

**Termina Condición 3.**

**Condición 4.** Verifica el método Runge-Kutta a utilizar.

Mueve las partículas (advección + difusión-turbulenta).

**Termina Condición 4.**

**Condición 5.** Si se desea considerar algún decaimiento.

Elimina partículas.

**Termina Condición 5.**

**Condición 6.** Si se desean visualizar y/o guardar imágenes de estadísticas.

Calcula las distintas cantidades de petróleo, y gráficaslas.

**Termina Condición 6.**

**Condición 7.** Si se desean guardar los datos numéricos de las trayectorias.

Guarda la información de las partículas en archivos .mat.

**Termina Condición 7.**

**Termina Ciclo 2.**

**Termina Ciclo 1.**