

# *Haml*: Mecánica hamiltoniana en un plano

Amilcar Meneses Viveros\*  
Sección de Computación  
Departamento de Ingeniería Eléctrica  
CINVESTAV-IPN

Agosto 2004

## Resumen

El programa *Haml* permite realizar la simulación del movimiento de una partícula en un plano, utilizando mecánica hamiltoniana. En este reporte se presenta la implantación y guía de uso de la versión para Mac OS X; y algunos comentarios sobre la simulación obtenida.

La aplicación *Haml* permite realizar la simulación del movimiento de una partícula en un plano, utilizando la mecánica hamiltoniana. *Haml* aprovecha las ventajas de la interfaz gráfica de usuario de Mac OS X, y del manejo de los objetos de la biblioteca de Cocoa para tener un buen desempeño y presentación de la simulación.

En este documento se presenta la implantación de *Haml* en la plataforma de desarrollo de Cocoa, utilizando como base el diseño desarrollado para NeXSTEP [3], en el Departamento de Aplicación de Microcomputadoras del Instituto de Ciencias de la UAP, sin embargo, se le han hecho algunas modificaciones para aumentar el rendimiento en la presentación gráfica.

Un conocimiento básico de mecánica clásica y las ecuaciones de Hamilton se necesitan para entender el funcionamiento de *Haml*, esto se discute en la primera sección; el caso de estudio que se presenta en *Haml* se discute en la sección 2. Posteriormente se discute el manejo de la aplicación. La sección 4 presenta el diseño de la aplicación, y de algunas estrategias en la implantación que mejoran su desempeño. Finalmente se presentan las conclusiones y perspectivas futuras de este trabajo.

Adicional a la sección “*Guía de usuario*” (sección 3), se ha agregado un apéndice donde se simplifican las acciones que el usuario puede hacer en *Haml*.

---

\*realizando estancia doctoral en el DIAS

## 1 Ecuaciones de movimiento de Hamilton

Un conocimiento básico de las ecuaciones de Hamilton para mecánica clásica es suficiente para trabajar con *HamI*. En esta sección se presentan las ecuaciones del movimiento de Hamilton. Una presentación más detallada se puede encontrar en [1],[2].

En mecánica clásica, el estado dinámico de un sistema puede definirse por sus coordenadas generalizadas  $q_1, q_2, \dots, q_R$  y sus momentos conjugados correspondientes  $p_1, p_2, \dots, p_R$ . Se introduce entonces un espacio de  $2R$  dimensiones: *el espacio fase*, en el que un estado dinámico está representado por un punto  $P$  cuyas coordenadas son las  $q$  y las  $p$ . La función de Hamilton se define como:

$$H \equiv H(q_1, q_2, \dots, q_R; p_1, p_2, \dots, p_R; t) = \sum_{r=1}^R \dot{q}_r \frac{\partial L}{\partial \dot{q}_r} - L$$

donde  $L$  es la función de Lagrange.

Las ecuaciones del movimiento de Hamilton se definen como

$$\dot{q}_r = \frac{\partial H}{\partial p_r} \quad r = 1, 2, \dots, R \quad (1)$$

$$\dot{p}_r = -\frac{\partial H}{\partial q_r} \quad r = 1, 2, \dots, R \quad (2)$$

Estas constituyen un conjunto de ecuaciones de primer orden acopladas. Al conocer las coordenadas y momentos iniciales del sistema, las ecuaciones de Hamilton (1, 2) permiten determinar su valor en los instantes posteriores.

## 2 Movimiento de una partícula sobre un campo magnético

Las corriente eléctricas producen campos magnéticos. La intensidad y dirección de un campo magnético se representa por  $\mathbf{B}$ .

Un campo magnético y un campo eléctrico en el vacío se pueden expresar como en [1]:

$$\mathbf{B} = \nabla \times \mathbf{A} \quad (3)$$

$$\mathbf{E} = -\nabla\phi - \frac{1}{c} \frac{\partial \mathbf{A}}{\partial t} \quad (4)$$

donde  $\mathbf{A}$  es el potencial vectorial,  $\phi$  es el potencial escalar y  $c$  es la velocidad de la luz.

El hamiltoniano de un sistema con una partícula con carga  $e$  y un potencial vectorial  $\mathbf{A}$  es

$$H(\mathbf{q}, \mathbf{p}, t) = \frac{1}{2m} \left[ \mathbf{p} - \frac{e}{c} \mathbf{A}(\mathbf{q}, t) \right]^2 + e\phi(\mathbf{q}, t)$$

donde  $\mathbf{p}$  queda como

$$\mathbf{p} = \frac{\partial L}{\partial \dot{\mathbf{q}}} = m\dot{\mathbf{q}} + \frac{e}{c} \mathbf{A} = m\mathbf{v} + \frac{e}{c} \mathbf{A}$$

y las ecuaciones de Hamilton quedan:

$$\dot{\mathbf{q}} = \mathbf{v} = \frac{1}{m} \left( \mathbf{p} - \frac{e}{c} \mathbf{A} \right) \quad (5)$$

$$\dot{\mathbf{p}} = -\nabla H = -e\nabla\phi - \frac{e}{c} \nabla(\mathbf{v} \cdot \mathbf{A}) \quad (6)$$

## 2.1 Un ejemplo específico con un campo magnético no homogéneo

La simulación que actualmente se presenta en *HamI* se basa en el siguiente potencial vectorial

$$\mathbf{A} = (-\cos(2y), \cos(\pi \sin(x)), 0) \quad (7)$$

Así, el campo magnético resultante es:

$$\begin{aligned} \mathbf{B} &= \nabla \times \mathbf{A} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ -\cos(2y) & \cos(\pi \sin(x)) & 0 \end{vmatrix} \\ &= k \left( \frac{\partial}{\partial x} \cos(\pi \sin(x)) + \frac{\partial}{\partial y} \cos(2y) \right) \\ &= k(-\pi \sin(\pi \sin(x)) \cos(x) - 2 \sin(2y)) \end{aligned} \quad (8)$$

De la ecuación (8), es posible identificar al campo en función de las coordenadas cartesianas  $x, y$ , a lo largo del tercer eje,  $z = -\pi \sin(\pi \sin(x)) \cos(x) - 2 \sin(2y)$ . La superficie de este potencial se muestra en la figura 2.1.

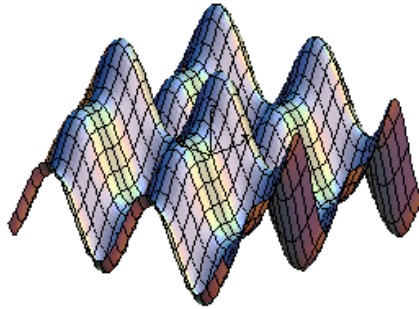


Figura 1: El campo como superficie

Los contornos de superficie del potencial se muestran en la gráfica (2.1). Los colores azules representan la parte negativa, el área blanca en estas zonas muestra la parte más negativa. Similarmente, los colores rosados representan la parte positiva del campo y, las áreas blancas en esta zona representa la parte más positiva.

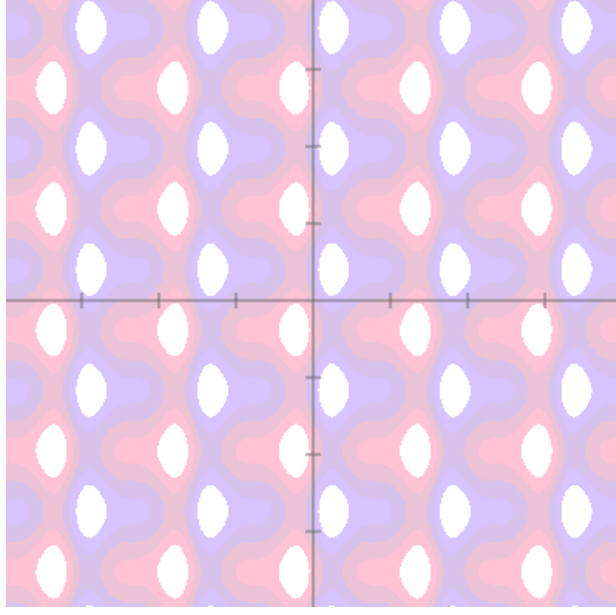


Figura 2: Contornos de superficie del campo

Si establecemos la masa de la partícula como 1, y denotamos  $\gamma = e/c$ , entonces

$$\mathbf{v} = \dot{\mathbf{q}} = (\mathbf{p} - \gamma \mathbf{A}) = (p_1 + \gamma \cos(2y), p_2 - \gamma \cos(\pi \sin x))$$

así

$$\dot{q}_1 = p_1 + \gamma \cos(2y) \quad (9)$$

$$\dot{q}_2 = p_2 - \gamma \cos(\pi \sin x) \quad (10)$$

al hacer cero el potencial escalar en (6) obtenemos

$$\begin{aligned} \dot{\mathbf{p}} &= \gamma \nabla(\dot{\mathbf{p}} \cdot \mathbf{A}) \\ &= \gamma \nabla(-\dot{q}_1 \cos(2y) + \dot{q}_2 \cos(\pi \sin x)) \\ &= \gamma \left[ \frac{\partial(-\dot{q}_1 \cos(2y) + \dot{q}_2 \cos(\pi \sin x))}{\partial x}, \frac{\partial(-\dot{q}_1 \cos(2y) + \dot{q}_2 \cos(\pi \sin x))}{\partial y} \right] \end{aligned} \quad (11)$$

Desarrollando con elemento  $p_1, p_2$  de esta ecuación tenemos

$$\dot{p}_1 = \gamma \pi \sin(\pi \sin x) \cos x (-\dot{q}_2 + \gamma \cos(\pi \sin x)) \quad (12)$$

$$\dot{p}_2 = \gamma 2 \sin(2y) (\dot{q}_1 + \gamma \cos(2y)) \quad (13)$$

Las ecuaciones (9, 10, 12, 13) las identificamos con las ecuaciones de Hamilton para nuestro ejemplo.

## 2.2 El caso homogéneo

Se tiene el caso homogéneo cuando se considera constante al potencial vectorial.

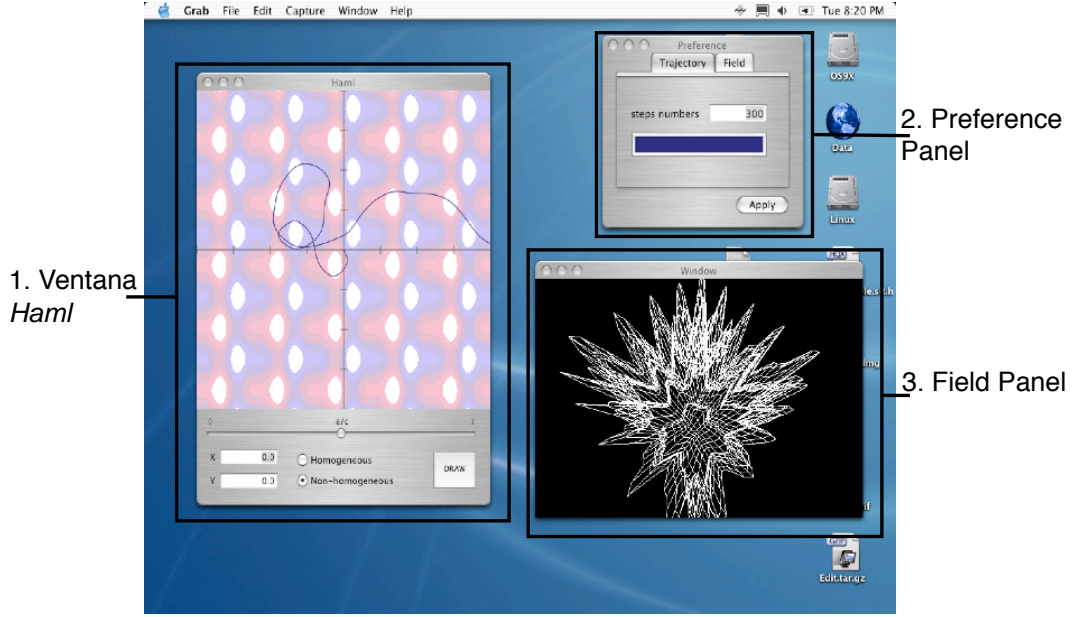


Figura 3: Entorno de la aplicacion *HamI*

Las ecuaciones de Hamilton se tienen como

$$\dot{q}_1 = p_1 + \gamma y \quad (14)$$

$$\dot{q}_2 = p_2 - \gamma x \quad (15)$$

$$\dot{p}_1 = \gamma \dot{q}_2 - \gamma^2 x \quad (16)$$

$$\dot{p}_2 = -\gamma \dot{q}_1 - \gamma^2 y \quad (17)$$

### 3 Guía de uso

*HamI* es una aplicación que muestra la simulación del movimiento de una partícula en un campo magnético no homogéneo.

*HamI* es una “Cocoa Application”, que corre en la plataforma Mac OS X. En la figura (3), se aprecia el total de elementos graficos que componen a la aplicación.

1. *Ventana HamI*. Es la ventana principal de la aplicación. Aquí se despliega la simulación y se dan los parámetros iniciales (origen y el valor del parámetro  $\gamma = e/c$ ).
2. *Panel de preferencias*. En este panel se guardan los colores con los cuales se trazan la trayectoria de la partícula y con los que se representan los diferentes contornos del campo magnético.
3. *Ventana del campo*. En esta ventana se despliega la superficie del campo magnético.

Con estos elementos, la aplicación es capaz de presentar la simulación requerida, teniendo en cuenta los parámetros de entrada, y algunos elementos auxiliares gráficos que le permiten al usuario para visualizar e interpretar el resultado.

### 3.1 La ventana *Haml*

La ventana principal de *Haml* permite al usuario dar los parámetros de entrada: posición inicial y el valor  $\gamma = e/c$ ; elegir el tipo de campo electromagnético, y observar el resultado de la simulación. Esta ventana se puede apreciar en la figura (4).

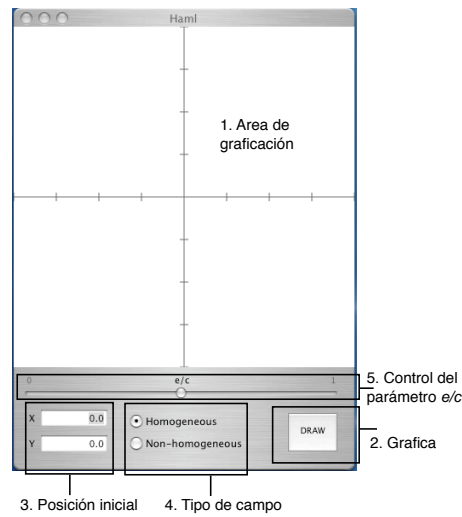


Figura 4: Ventana principal de *Haml*

1. *Area de graficado*. En esta área se despliega el resultado de la simulación. La simulación de la trayectoria que sigue la partícula se representa sobre el plano  $XY$ , el eje  $X$  es el horizontal y, por ende, el  $Y$  es el vertical. El área que se presenta en el plano

$$XY = \{(x, y) | x, y \in (-7.97, 7.97)\}$$

Si se ha dado la opción de campo no homogéneo, el fondo permanecera en blanco, si el fondo se ha seleccionado homogéneo, entonces se desplegarán los contornos de superficie del campo magnético. El color de la trayectoria y de los contornos se define en el *Panel de Preferencias*. La posición inicial se define en la *Ventana Haml*. El valor del parámetro  $\gamma = e/c$  se proporciona en la *Ventana Haml*.

2. *Botón Grafica (Draw)*. Este botón genera que se despliegue el resultado de la simulación, dependiendo de los valores iniciales, tipo de campo, y de los colores que se tengan seleccionados.
3. *Posición Inicial*. Este campo permite dar la posición inicial de la simulación, o visualizar los valores correspondientes cuando la selección se ha realizado a través del ratón (ver apéndice A, “*Acciones del usuario: Indicando la posición inicial*”).

4. *Tipo de Campo*. Este conjunto de botones indica si la simulación se realizará con el campo homogéneo, o con un campo constante (caso no homogéneo).
5. *Control del parámetro  $\gamma = e/c$* . El parámetro  $\gamma$  se controla a través de una palanca, el rango que cubre este valor es  $\gamma \in [0, 1]$ . Una característica atractiva es que el usuario puede cambiar este valor y ver directamente el resultado de la simulación, sin necesidad de accionar el botón *Grafica* (Draw) de la ventana *Haml*.

### 3.2 El panel de preferencias

Este panel maneja las opciones de preferencias del usuario. Las opciones que el usuario puede

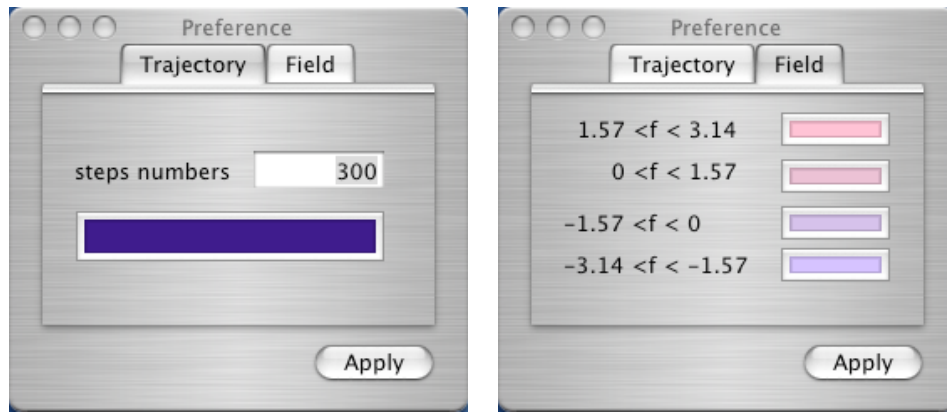


Figura 5: Opciones del panel de preferencias del usuario

guardar a través de distintas ejecuciones de *Haml* son el número de pasos que tendrá la trayectoria de la partícula, y el conjunto de colores con los que se representa el movimiento de la partícula, y los contornos del campo magnético. Esto es, una vez que se seleccionan y se presiona el botón de aplicar (*Apply*). Los colores se respetan la próxima vez que se ejecute la aplicación.

El panel de preferencias (*“Preferences”*) tiene dos grupos de opciones principales: *Trajectory* y *Field*.

- Opción *Trajectory*. Permite manejar las preferencias del usuario respecto a los atributos de la trayectoria de la partícula, estos atributos son el número de pasos de la trayectoria y el color con el que se presentará.
- Opción *Field*. Permite asignar un color a un rango del valor de la superficie para la representación de los contornos. Hay cuatro niveles que cubren los rangos  $(-3.14, -1.57)$ ,  $(-1.57, 0)$ ,  $(0, 1.57)$  y  $(1.57, 3.14)$ . Los rangos mínimos y máximos se seleccionaron hasta 3.14 y -3.14, respectivamente, debido a la forma de la función que representa el campo, como se aprecia en la ecuación (8).

### 3.3 La ventana del campo

La ventana del campo permite al usuario tener una visualización más directa de la función que representa el campo, ecuación (8). La característica principal de esta ventana es la incorporación de “*trackball*” para visualizar a la superficie. El usuario puede rotar la imagen con la acción de arrastrado del ratón en sobre el área de la ventana.

Si el usuario cuenta con un ratón de dos botones puede hacer una translación con la acción de arrastrado, utilizando el botón derecho, (la translación se efectúa sobre el plano de proyección de la imagen de la superficie); si el usuario no cuenta con este tipo de ratón, se puede tener este comportamiento usando la secuencia <CTRL>-<DRAG>, es decir, presionar la tecla <CTR> y hacer la acción de arrastre sobre la superficie de la ventana.

La última operación característica en esta ventana es el escalamiento de la imagen. Si el usuario cuenta con un ratón con un “*scroll*”, lo puede utilizar para esta operación. Este mismo efecto se tiene utilizando la secuencia <ALT>-<DRAG>, es decir, presionar la tecla <ALT> y realizar la acción de arrastre del ratón sobre la superficie de la ventana.

Las acciones que el usuario puede realizar en *Haml* estan sumariadas en el apéndice A.

## 4 La implantación

El núcleo *Haml* fue desarrollado por el Dr. Harold V. McIntosh en 1995 y en 1996 se integra como una aplicación de NeXTSTEP [3]. Esta aplicación consistía de una ventana principal donde se presentaba la simulación y se desplegaba el campo como superficie o como contorno. Para esta última tarea se utilizó la versión para NeXTSTEP del paquete PLOT —PLOT es un paquete gráfico desarrollado por el Dr. Harold V. McIntosh en Salazar, y posteriormente se ha migrado a diferentes plataformas en el Departamento de Aplicación de Micromputadoras del ICUAP—.

Hay dos maneras generales de migrar aplicaciones de NeXTSTEP a Mac OS X. La primera es pasar la aplicación a OpenStep, como paso intermedio, utilizando los *scripts* disponible en el ambiente OpenStep, y después realizar la migración a Mac OS X, con ayuda de los *scripts* disponibles en esta última plataforma. La otra opción es tomar el modelo orientado a objetos original de la aplicación, y reescribirla directamente, tomando en cuentas las características técnicas de la plataforma donde se migrará la aplicación. En este caso se optó esta última opción, y se ha tratado de mantener el diseño original.

Las aplicaciones de Cocoa han cambiado de las aplicaciones de NeXTSTEP, aunque mucho de la filosofía de trabajo de NeXTSTEP quedó vigente en Mac OS X, también han cambiado muchos aspectos técnicos. Tal vez los más importantes son el manejo de video, ya que en NeXTSTEP se realizaba con operadores de PostScript (con el concepto de Display PostScript), mientras que en Cocoa (la versión de OpenStep para mac OS X) la salida gráfica se realiza por medio de los objetos *NSBezierPath*. Otro cambio importante es cambio de las bibliotecas de objetos, ya que en OS



X se trabaja con la estandarización del API<sup>1</sup> de `OpenStep`. Además, Mac OS X incorpora en su plataforma tecnologías como OpenGL, y que pueden incorporarse a las aplicaciones de Cocoa.

Para la versión en Mac OS X se ha tratado de conservar el diseño que se desarrollo para NeXTSTEP, pero incorporando los aspectos técnicos de la nueva plataforma de trabajo y el uso de las tecnologías disponibles.

El manejo de la aplicación y, principalmente la interfaz gráfica de usuario se manejan desde Cocoa con Objective C, y una parte de la salida con OpenGL.

El núcleo principal de *Haml* se ha mantenido en C por fines de rendimiento y portabilidad a otras plataformas. Esto evita la sobrecarga que se pudiera obtener al utilizar el manejo de métodos de objetos programados en Objective C, en lugar de funciones.

*Haml* utiliza el método Runge-Kutta de cuarto orden para resolver numéricamente las ecuaciones (1)(2), una discusión más detallada sobre este método se puede encontrar en [9], [10], [11]. Las condiciones iniciales están determinadas por el punto donde la partícula iniciará su trayectoria.

Como se ha mencionado, *Haml* conserva el núcleo escrito en C, este se encuentra en el archivo `haml.c` y cuenta principalmente con las funciones para realizar el cálculo del gradiente —que es la función que se utiliza para calcular el valor de la derivada de  $\dot{\mathbf{p}}$ , como se muestra en la ecuación (11)—, la rutina de integración Runge-Kutta de cuarto orden, y algunas rutinas auxiliares.

Respecto al diseño orientado a objetos, *Haml* cuenta con tres clases principales: `HamController`, `HamView` y `FieldGLView`. La primera es el controlador de la aplicación, la segunda se encarga de desplegar el resultado gráfico de la simulación y la tercera se encarga de mostrar la superficie del campo con ayuda de OpenGL.

La clase `HamController`, se utiliza para labores de inicialización de la aplicación y del manejo del panel de preferencias del usuario. Por propósitos de eficiencia, esta clase se encarga de crear un objeto `NSThread` para calcular la matriz de superficie del campo magnético, que después utilizarán los objetos `HamView` y `FieldGLView` para mostrar los contorno de superficie y la superficie del campo, respectivamente. Esto permite que el programa no tenga que calcular esta superficie cada vez que se desee volver a desplegar alguno de los objetos que muestran el campo. Otra tarea es el manejo de las preferencias del usuario, esto se realiza manejando el archivo con formato *Property List*, que utilizan las aplicaciones de Cocoa para esta tarea. Una discusión más detallada sobre las aplicaciones Cocoa y el manejo de características como la configuración de la aplicación se puede encontrar en [5].

La clase `HamView`, se encarga de desplegar el resultado de la simulación, principalmente esta clase se auxilia de un objeto `NSImage` donde se dibuja el contorno del campo. La imagen del objeto `NSImage` se construye por un hilo cuando se termina de cargar el módulo *NIB*<sup>2</sup> que contiene este objeto. Este hilo utiliza la matriz que previamente a llenado otro hilo mandado a ejecutar desde el objeto `HamController` (evidentemente se utiliza un candado para evitar que este hilo lea una matriz con basura, en caso de que el hilo en `HamController` no la haya terminado de llenar). Esta estrategia

---

<sup>1</sup>API:Application Program Interface

<sup>2</sup>NIB: módulo del *Interface Builder*.

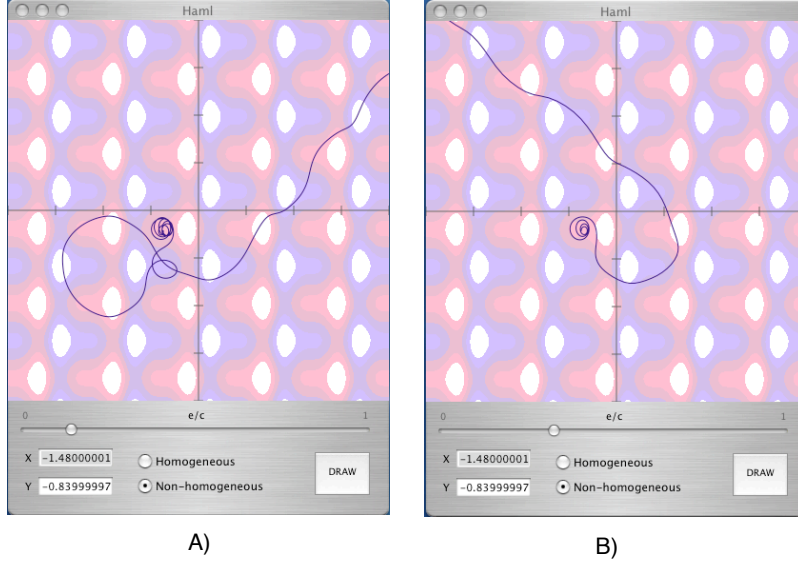


Figura 6: Variación del parámetro  $\gamma$  con la misma posición inicial

permite que no se tenga que redibujar el campo cada vez que se presente una nueva atrayectoria. Esto es, cuando al objeto se le manda el mensaje para mostrar la gráfica de la simulación, lo primero que hace es calcular la nueva trayectoria llamando al núcleo de *HamI*, después forma un objeto `NSBezierPath` (para que se pueda desplegar la trayectoria), se manda a desplegar el objeto `NSImage`, con el dibujo del campo como imagen de fondo, y finalmente, se sobreponen los ejes y la trayectoria contenida en el objeto `NSBezierPath`.

La clase `FieldGLView` se encarga de desplegar la superficie del campo. Esta clase permite incorporar el ambiente de OpenGL. Para poder controlar operaciones de rotación, translación y escalamiento de la superficie con el ratón se utilizó como base el código `trackball` del código de ejemplo `Cocoa OpenGL`, distribuido por Apple Inc. [8]. Sin embargo se procuro que la aplicación fuera independiente de `Core`, ya que la mayoría de los ejemplos que ofrece Apple mantienen la incorporación de las bibliotecas `Core`. Una mayor discusión sobre OpenGL y su uso en el ambiente Mac OS X se puede obtener en [7] y [6].

## 5 Comentarios sobre la simulación

En mecánica clásica el movimiento de una partícula se puede determinar por la acción del campo magnético que actúa sobre ella. Esto se refleja en la trayectoria resultante en la simulación. En esta sección veremos algunas características del caso no-homogéneo y posteriormente se mostrarán los resultados del caso homogéneo.

## 5.1 El caso no-homogéneo

Se ha seleccionado un potencial vectorial (7) que produzca un campo magnético no homogéneo (8) con características periódicas, como se puede apreciar en las gráficas de superficie (2.1) y contornos (2.1). Esta periodicidad proporciona una especie de “regularidad” del campo nos permite ver algunas características generales del comportamiento de las partículas en este tipo de campos.

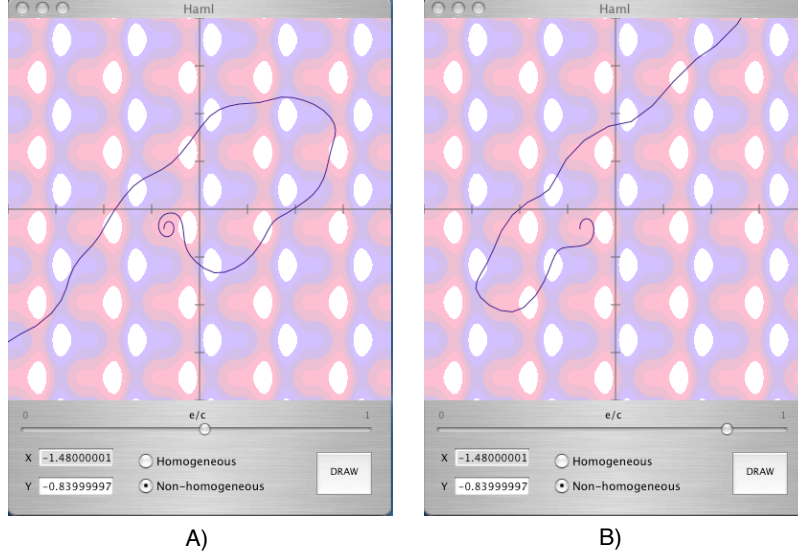


Figura 7: Variación del parámetro  $\gamma$  con la misma posición inicial

Por propósitos de la simulación los valores iniciales para  $\dot{p}_1$  y  $\dot{p}_2$  se han escogido como  $\dot{p}_1 = 0.25$  y  $\dot{p}_2 = -2.0$ , en versiones posteriores de *HamI*, este parámetro podrá ser manipulado por el usuario.

Dependiendo del valor del parámetro  $\gamma = e/c$ , la trayectoria puede quedar atrapada por un lapso mayor de tiempo antes de escapar a otra región, esto se aprecia en las figuras (6)(7). A mayor intensidad del campo, es decir, mientras  $\gamma$  se incrementa, escapará más rápido de una región. La tabla (1) muestra la relación de las figuras con el valor de  $\gamma$ :

Figura	valor de $\gamma$
6.A	.12
6.B	.29
7.A	.51
7.B	.79

Tabla 1: Valores de  $\gamma$  en las figuras (6)(7)

En las figuras (6)(7), también se puede observar que cuando una partícula sale de una región positiva (región rosada) avanza a una región donde el campo tiene carga negativa (región azul) debido a que la partícula ha adquirido una carga positiva y es rechazada por el mismo campo, y así, es atraída por la región del campo con carga contraria. A su vez, la partícula empieza a

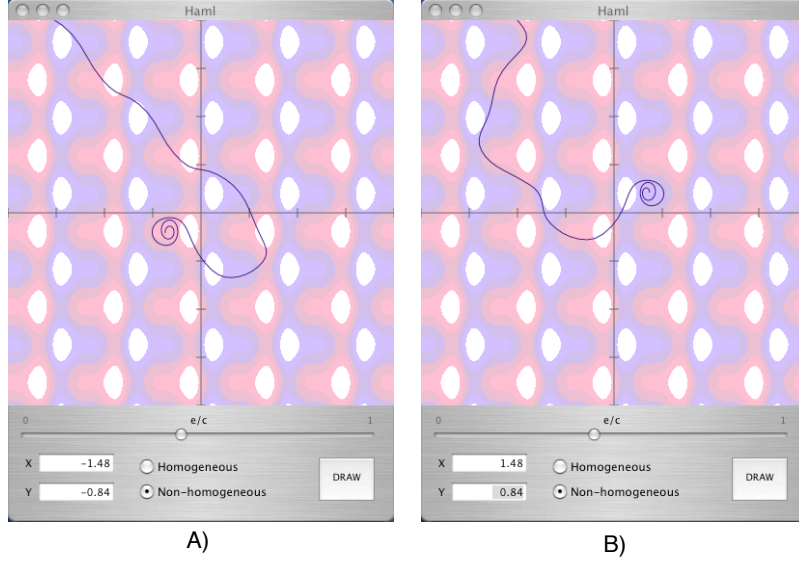


Figura 8: Simulación partiendo de una zona con carga positiva (A) y carga negativa (B)

cambiar de carga y cuando adquiere una carga negativa, sale de esta región hacia una zona del campo con carga positiva. Por el tipo de campo periódico del ejemplo, este comportamiento se repite constantemente.

Es posible ver el cambio en la dirección de la trayectoria al entrar en una zona del campo con carga positiva o negativa. En la figura (8) se aprecia con más detalle este efecto al colocar la posición inicial de la partícula en una zona con carga negativa (8.A) y en una zona con carga positiva (8.B). La partícula que inicia en zona positiva tiene un desplazamiento en el sentido de las manecillas del reloj. Mientras que, la partícula que inicia su trayectoria en una zona del campo con carga negativa tiene un desplazamiento contrario al de las manecillas del reloj.

## 5.2 El caso homogéneo

Por fines de comparación se incluyó en *HamI* el caso homogéneo. En este caso el potencial es constante y el campo electromagnético resultante puede pensarse, en términos de superficie, como un plano paralelo a una distancia del plano  $YX$  donde se mueve la partícula. El valor  $\gamma = e/c$  se comporta como un factor de escala sobre el eje de proyección  $Z$ , es decir, indica que tan cercano o alejado está el campo electromagnético del plano  $XY$ .

La figura (6) muestra el comportamiento de la partícula para diferentes valores de  $\gamma = e/c$ . El inicio de la trayectoria se ha fijado en las coordenadas  $(0, 0)$ , se puede apreciar que la trayectoria se realiza conforme a las manecillas del reloj, ya que el campo se encuentra en una posición “positiva” sobre el eje de proyección  $Z$ .

Evidentemente el comportamiento varía si se toman otros puntos iniciales. En la figura (10). Se puede apreciar que las distintas trayectorias son en dirección de las manecillas del reloj y al rededor

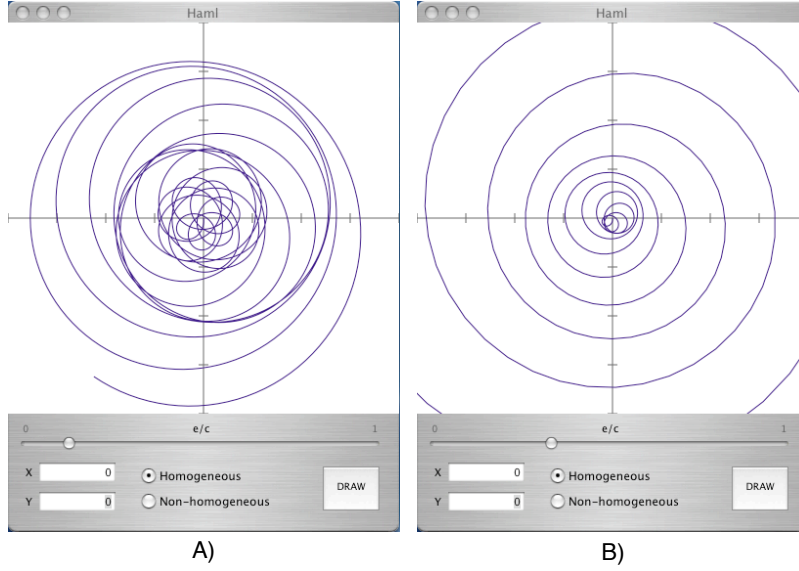


Figura 9: El caso homogéneo, variando  $\gamma$

del origen. De hecho, este comportamiento también se tiene en el caso mostrado en la figura (9).

## 6 Resultados y conclusiones

Iniciaremos esta sección con algunas discusiones técnicas. Se ha logrado migrar la aplicación *Haml*, conservando el núcleo del programa, haciendo la implantación con base al modelo original y aprovechando algunas características de la plataforma de Mac OS X. La estrategia de multihilos y la combinación de los objetos `NSImage` y `NSBezierPath` aumentan considerablemente el desempeño de la aplicación, esto permite que los resultados se presenten dinámicamente cuando el usuario cambia algún valor. Particularmente la manera de utilizar el “trackball” y el ambiente OpenGL, nos permite visualizar algunas características que se pueden agregar al paquete de graficación *PLOT*, con el se tiene implantado *Haml* en NeXTSTEP.

Por otro lado, una parte que resulta interesante es la simulación de la partícula para diferentes campos que presenten un comportamiento no homogéneo. Debido a que la forma de el campo electromagnético  $\mathbf{B}$  se puede obtener a partir del potencial vectorial  $\mathbf{A}$ , tal y como se aprecia en la ecuación (8). El campo se puede programar a partir del un campo vectorial  $\mathbf{A}$ . Es posible agregar a *HAML* la parte de un analizador sintáctico para obtener el campo partiendo de un potencial vectorial, el cual se obtendría desde la interfaz de usuario.

Se tiene previsto agregar algunas facilidades al usuario, como son el manejo del área del plano  $XY$  que se desplegará; la incorporación de otros valores como son momento inicial y la masa; y la cantidad rangos para superficie de nivel y su manejo de color respectivo.

Finalmente, podemos decir que con algún conocimiento básico de sistemas físicos, y el manejo

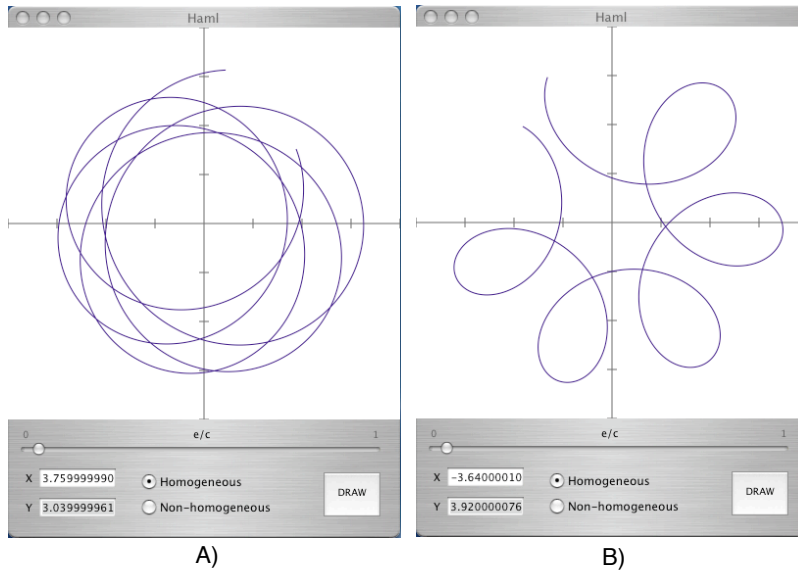


Figura 10: Caso homogéneo con diferentes puntos iniciales

de estrategias de programación hacen posible el desarrollo de aplicaciones que permiten simular fácilmente algunos fenómenos físicos con buen desempeño.

## 7 Comentarios finales sobre la aplicación

A excepción del núcleo de *HamI*, la aplicación fue reescrita totalmente, tomando como base el diseño que se utilizó para la versión de NeXTSTEP desarrollada en el Depto. de Aplicación de Microcomputadoras del ICUAP [3] —esto se aprecia, además del diseño que contaba con el despliegado de la trayectoria sobre el campo, y una ventana especial para mostrara la superficie del campo; y en el ejemplo del potencial que se utiliza para presentar la simulación—. Los cambios significativos que se le realizaron fueron: evidentemente el cambio de ambiente de NeXTSTEP a OS X; la incorporación de OpenGL para mostrar la superficie del campo; incorporación del panel de preferencias del usuario; la combinación de objetos `NSImagen` y `NSBezierPath` de la biblioteca de Cocoa [4] para presentar dinámicamente los resultados a los eventos del usuario; y finalmente, la estrategia de multihilos para calcular la superficie del campo mientras se inicializa la aplicación.

La aplicación ejecutable y el código fuentes se pueden obtener de

[http://computacion.cs.cinvestav.mx/~ameneses/~amv\\_proy.html#haml](http://computacion.cs.cinvestav.mx/~ameneses/~amv_proy.html#haml)

Los requerimientos para el código ejecutable es tener al menos la version 10.2 de Mac OS X.

## A Acciones del usuario

**Seleccionando tipo de campo.** Los botones “Tipo de campo” en la ventana principal permite realizar esta selección. Hasta el momento se ha colocado un solo tipo de campo no-homogéneo, cuya descripción se ha realizado en la sección 3.

**Indicando la posición Inicial.** Hay dos maneras de hacer esta tarea. La primera es escribiendo los valores de  $x$  y  $y$  en los campos de texto asociados a estos valores en la ventana *Haml* —recuerde, el área que se presenta en el plano  $XY = \{(x, y) | x, y \in (-7.97, 7.97)\}$ —. La otra opción es haciendo un “click” (presionar el botón derecho del ratón) en el *area de graficación*, de la ventana *Haml*, donde se quiera la posición inicial de la partícula.

**Cambiando el valor de  $\gamma = e/c$ .** Mover, con el ratón, la palanca asociada al valor  $\gamma$  en la ventana *Haml*. Automáticamente se desplegará el resultado de la simulación.

**Cambiar el número de pasos de la trayectoria.** Desplegar el panel de preferencias, seleccionar el grupo de opciones *Trajectory* y cambiar el número de pasos. Finalmente, presionar el botón *Apply*, para que la aplicación tenga presente el cambio.

**Cambiando el color de la trayectoria.** Desplegar el panel de preferencias, seleccionar el grupo de opciones *Trajectory* y cambiar el color del pozo haciendo un “click” sobre el marco del pozo de color, con lo que se desplegará el panel de colores, seleccionar el que se quiera. Finalmente, presionar el botón *Apply*, para que la aplicación tenga presente el cambio.

**Cambiando el color de los contornos del campo.** Desplegar el panel de preferencias, seleccionar el grupo de opciones *Field* y cambiar el color del pozo asociado al nivel de contorno que se desee, haciendo un “click” sobre el marco del pozo de color, con lo que se desplegará el panel de colores, seleccionar el que se quiera. Finalmente, presionar el botón *Apply*, para que la aplicación tenga presente el cambio.

**Desplegar la ventana del campo.** Seleccionar el submenú *Tools* del menú principal de *Haml*, y después seleccionar la opción *Show Field*.

**Desplegar el panel de preferencias.** Seleccionar el submenú *Haml* del menú principal de la aplicación, y después seleccionar la opción *Preferences*. Esto también se logra con la secuencia de teclas <APPL>+<,>.

**Rotar la superficie del campo.** Desplegar la ventana del campo, después realizar una operación de arrastre sobre el área de esta ventana. Un arrastre vertical producirá una rotación sobre el eje  $Y$  de proyección, y un arrastre horizontal producirá una rotación sobre el eje  $X$  de proyección.

**Cambiar la escala de la superficie del campo.** Se puede realizar de dos formas. La primera es si se cuenta con un ratón con “scroll”; el “scroll” utiliza para esta operación, siempre y cuando el apuntador del ratón esté sobre el área de la ventana del campo. La segunda opción es utilizar la secuencia <ALT>-<DRAG>, es decir, presionar la tecla <ALT> y realizar la acción de arrastre del ratón sobre la superficie de la ventana.

**Transladar la superficie del campo.** Se puede realizar de dos formas. La primera es si se cuenta con un ratón con dos botones, entonces se presiona el botón derecho sobre el área

de la ventana y se realiza la operación de arrastre. La segunda opción es utilizar la secuencia <CTRL>-<DRAG>, es decir, precionar la tecla <CTRL> y hacer la operación de arrastre de ratón sobre el área de la ventana del campo.

## Referencias

- [1] H.C. Corben, Philip Sthele; *“Classical Mechanics”*; John Wiley & Sons, Inc.; New York; 1950.
- [2] H. Weyl; *“The Theory of Groups and Quantum Mechanics”*; Methuen & Co. Ltd; London, 1931.
- [3] J. P. Hernández Enríquez, A. Meneses Viveros, M. Roldán Palacios y S. Zárate Vázquez; *“Hamiltoniano”*; Reporte final del curso de Análisis Numerico III; Jun-1997; Depto. de Aplicación de Microcomputadoras, ICUAP, Puebla, México.
- [4] Apple Inc. *Cocoa Reference Library*, <http://developer.apple.com/referencelibrary/Cocoa/>, 2003.
- [5] Apple Inc. *“Inside Mac OS X: System Overview”*; <http://developer.apple.com/documentation/MacOSX/Conceptual/SystemOverview/>; 2002.
- [6] Apple Inc. *“Inside Mac OS X: System Overview”*; <http://developer.apple.com/documentation/Cocoa/Conceptual/OpenGL/>; 2003.
- [7] M. Woo, J. Neider, & T. Davis; *“OpenGL, Programming Guide”*; Addison-Wesley Developer Press, 1997.
- [8] Apple Inc. *“Cocoa OpenGL”*; [http://developer.apple.com/samplecode/Cocoa\\_OpenGL/Cocoa\\_OpenGL.html](http://developer.apple.com/samplecode/Cocoa_OpenGL/Cocoa_OpenGL.html); 2003.
- [9] Lapidus and Seinfeld; *“Numerical Solutions of Ordinary Differential Ecuations”*; Academic Press; New York, 1963.
- [10] B. Carnahan, J.O. Wilkes; *“Digital Computing and Numerical Methods”*; John Wiley & Sons, New York; 1973.
- [11] W.H. Press, S.A. Teukolsky, W.T. Vetterling, & B.P. Flannery; *“Numerica Recipes in C; The Art of Scientific Computing”*; Second Edition, Cambridge University Press; New York; 1992.