

# PRÁCTICAS DE CÁLCULO NUMÉRICO

## PRÁCTICA 4

Página web de la asignatura: <http://personales.unican.es/segurajj/docencia.html>

El objetivo de esta última práctica de la asignatura es resolver un problema de Mecánica Cuántica: la obtención de niveles de energía de un oscilador armónico perturbado. En particular, consideraremos, entre otras, una perturbación de tipo  $\lambda x^4$  (ver, por ejemplo, Galindo-Pascual [1]). El desarrollo de esta práctica requerirá implementar y analizar diferentes aspectos numéricos estudiados en la última parte del curso. En concreto, consideraremos cuadraturas numéricas (regla trapezoidal recurrente) para evaluar los elementos de matriz del hamiltoniano perturbado y aplicaremos el método de iteración inversa (método de la potencia inversa) para obtener la energía del estado fundamental perturbado y de los primeros niveles excitados.

### 1. El problema

El planteamiento del problema puede encontrarse en la página 167 de la referencia [1]. De forma resumida, consideramos el problema de un oscilador con hamiltoniano

$$\mathcal{H} = \mathcal{H}_0 + \mathcal{H}_1$$

donde  $\mathcal{H}_0 = -\frac{d^2}{dx^2} + x^2$  y  $\mathcal{H}_1$  es una perturbación; como en [1], hemos fijado las constantes a 1 para simplificar las expresiones (siempre se pueden recuperar las constantes a posteriori). Uno de los ejemplos de Hamiltoniano perturbado que consideraremos será  $\mathcal{H}_1(x) = \lambda x^4$ .

Los autovalores y autovectores (ortonormales) de  $\mathcal{H}_0$  son de sobras conocidos:

$$E_n^{(0)} = 2n + 1, \quad \Phi_n^{(0)}(x) = \frac{1}{(\sqrt{\pi}2^n n!)^{1/2}} H_n(x) e^{-x^2/2} \quad (1)$$

siendo  $H_n(x)$  el polinomio de Hermite de orden  $n$ .

La presencia del término de perturbación modificará los autoestados y sus correspondientes niveles energéticos. Por ejemplo, para el caso de la perturbación  $\lambda x^4$  sabemos que la energía del estado fundamental en teoría de perturbaciones viene dado, para  $\lambda$  “suficientemente pequeño”, por [1]:

$$E_0(\lambda) = 1 + \frac{3}{4}\lambda - \frac{21}{16}\lambda^2 + \mathcal{O}(\lambda^3) \quad (2)$$

Compararemos las predicción de la teoría de perturbaciones con los valores que se obtienen mediante la diagonalización numérica de la matriz hamiltoniana truncada.

El desarrollo de la práctica tendrá, de forma resumida, dos fases:

- a) Construcción de la matriz (truncada) asociada al hamiltoniano del sistema.
- b) Diagonalización de la matriz anterior.

Veamos qué elementos son necesarios en cada una de estas fases.

## 2. Desarrollo de la práctica:

### 2.1. Construcción de la matriz asociada al hamiltoniano

Los elementos de matriz a evaluar son:

$$h_{nm} = \langle \Phi_n^{(0)} | \mathcal{H} | \Phi_m^{(0)} \rangle = E_n^{(0)} \delta_{nm} + \mathcal{P}_{nm}, \quad (3)$$

donde  $\mathcal{P}_{nm} = \langle \Phi_n^{(0)} | \mathcal{H}_1 | \Phi_m^{(0)} \rangle$ .

Necesitamos entonces:

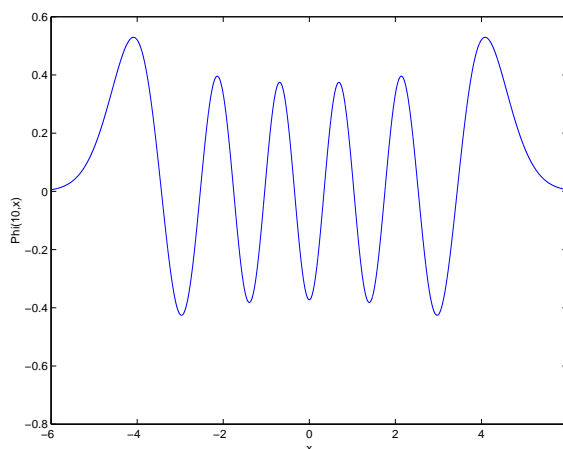
- 1) **Evaluar los autoestados del hamiltoniano no perturbado.** Para ello, utilizaremos la relación de recurrencia satisfecha por los polinomios de Hermite

$$H_{n+1} = 2xH_n - 2nH_{n-1}, \quad H_0 = 1, \quad H_1 = 2x.$$

lo que nos permite generar de forma estable y recursiva los polinomios de Hermite. El autoestado  $n$ -ésimo viene dado por la ecuación (1).

Específicamente: construiremos la rutina **hermitp**( $n, x$ ) cuya salida será el valor de la función  $\Phi_n^{(0)}(x)$  en  $x$ .

Como comprobación de que la función **hermitp** funciona correctamente, se sugiere dibujar las gráficas de estas funciones para ciertos valores del orden  $n$ , por ejemplo,  $n = 0, 1, 2, 5, 10$ . Se deben obtener funciones con un número de ceros igual a  $n$ . Por otra parte, las funciones  $\Phi_n(x)$  tienden a cero cuando  $x \rightarrow \pm\infty$ . Un ejemplo gráfico es el siguiente (se representa  $\Phi_{10}(x)$ ):



Otra comprobación la proporciona el cálculo de los ceros de estas funciones. Por ejemplo, los ceros de  $\Phi_2(x)$  son  $\pm 1/\sqrt{2}$  y los de  $\Phi_3(x)$  son  $\pm\sqrt{3/2}$  y 0 (ceros de los correspondientes polinomios de Hermite).

Conviene que nos aseguremos de que la función **hermitp** funciona correctamente puesto que los autoestados van a ser el punto de partida de todos los siguientes cálculos.

- 2) **Evaluar los elementos de matriz  $\mathcal{P}$ .** Es decir, tenemos que evaluar las integrales

$$\mathcal{P}_{nm} = \langle \Phi_n | \mathcal{H}_1 | \Phi_m \rangle = \int_{-\infty}^{\infty} \Phi_n(x) \Phi_m(x) \mathcal{H}_1(x) dx$$

Por supuesto, truncaremos la matriz (la matriz asociada al hamiltoniano es infinita).

Observemos que en el integrando, tenemos siempre un factor  $e^{-x^2}$  aportado por las autofunciones. Esto sugiere que la integración de Gauss-Hermite puede ser un método adecuado para calcular

estos elementos de matriz (sobre todo si  $\mathcal{H}_1$  es un polinomio en  $x$ ). Optaremos por un método más sencillo de implementar y que tiene un comportamiento más que bueno para este tipo de integrales con decaimiento rápido en  $\pm\infty$ : la regla trapezoidal recurrente. Para este tipo de integrales es de hecho la mejor opción entre las reglas de Newton-Cotes (o variantes como Romberg). Así pues, como paso previo, vamos a escribir un algoritmo que implemente la regla trapezoidal recurrente. Recordamos en este punto el algoritmo que describimos en clase:

**Algoritmo: Regla trapezoidal para evaluar  $\int_a^b f(x)dx$  con error absoluto menor que  $\epsilon$ .**

Input:  $\epsilon > 0$ ,  $f(x)$ ,  $b$ ,  $a$ .

Output:  $\int_a^b f(x)dx$

- (1)  $h = b - a$ ,  $I = \frac{h}{2}(f(a) + f(b))$
- (2)  $\Delta = 1 + \epsilon$ ;  $n = 0$
- (3) Repetir mientras  $\Delta > \epsilon$
- (4)  $n = n + 1$ ;  $h = h/2$
- (5)  $I_n = I/2 + h \sum_{i=1}^{2^{n-1}} f(a + (2i - 1)h)$
- (6)  $\Delta = |I_n - I|$
- (7)  $I = I_n$
- (6) Ir a (3)

Se trata de implementar este algoritmo en Matlab haciendo una serie de modificaciones. En primer lugar, consideraremos el criterio de error relativo en lugar del de error absoluto para parar el algoritmo. Esto quiere decir que tomaremos  $\Delta = |1 - I/I_n|$ . Esta modificación obligará a tomar alguna precaución adicional, como más adelante se comentará (por ejemplo, no se puede considerar el error relativo cuando  $I_n = 0$ ).

Para evitar problemas cuando no se alcance la precisión requerida conviene no tomar una precisión relativa demasiado exigente:  $10^{-10}$  es más que suficiente. A la vez, conviene también limitar el número máximo de iteraciones de la regla trapezoidal (10 iteraciones deben bastar)<sup>1</sup>. Por otro lado, conviene exigirle a la regla trapezoidal que realice un número mínimo de iteraciones (por ejemplo, 4 iteraciones al menos). En resumen, además del criterio de parada mediante el error relativo exigiremos que el algoritmo realice no menos de 4 y no más de 10 iteraciones.

Por supuesto, como las integrales que debemos calcular son impropias (se extienden desde  $-\infty$  a  $+\infty$ ) debemos limitar nuestros límites de integración de forma razonable. En la práctica, para las integrales que queremos calcular, será suficiente considerar la aproximación:

$$\int_{-\infty}^{+\infty} f(x)dx \approx \int_{-10}^{10} f(x)dx$$

Finalmente, como antes se adelantó, la utilización del error relativo en el criterio de parada obliga a tomar algunas precauciones para evitar errores cuando la integral que se evalúa es nula

<sup>1</sup>Conviene señalar que el máximo número de iteraciones que hemos adoptado (10 iteraciones), aún siendo pequeño, es una estimación conservadora. En realidad, la situación es incluso mejor: normalmente 7-8 iteraciones (64-128 evaluaciones) serán suficientes para alcanzar doble precisión. Para otro tipo de integrales, las cosas no tienen por qué funcionar tan bien pero para integrales impropias con integrandos analíticos (infinitamente derivables) y con un comportamiento como  $e^{-x^2}$  para  $|x|$  grande, esto es lo habitual.

o muy pequeña en módulo. Por ejemplo, cuando apliquemos nuestro programa al cálculo de  $\int_{-10}^{+10} xe^{-x^2} dx = 0$ , no se pueden considerar errores relativos en el criterio de parada. Debemos modificar el algoritmo para que se considere error absoluto en lugar de relativo cuando el valor absoluto de la integral sea muy pequeño (valores de  $|I_n|$  menores, por ejemplo, que  $\epsilon$ ).

En cualquier caso, como antes se comentó, no conviene aplicar ningún método de parada del algoritmo antes de que se hayan calculado al menos cuatro iteraciones de la regla trapezoidal recurrente.

*Específicamente: construiremos la rutina **traper**, cuya sintaxis de llamada será **traper(eps,a,b)** para calcular integrales  $I = \int_a^b f(x)dx$  con un error relativo **eps** (salvo cuando la integral se anule o sea muy pequeña en módulo, en cuyo caso **eps** será el error absoluto). La función  $f(x)$  se incluirá en un fichero que llamaremos **funci.m**.*

Para comprobar el funcionamiento de nuestra regla trapezoidal, se sugieren las siguientes integrales de prueba.

1.

$$\int_{-\infty}^{+\infty} \cos(nx)e^{-x^2} dx = \sqrt{\pi}e^{-n^2/4}$$

Considerar únicamente valores moderados de  $n$  ( $n \leq 8$ ) pues el cálculo numérico de esta integral se vuelve inestable a medida que crece  $n$  y no es posible obtener un error relativo de  $10^{-10}$  para  $n \geq 9$  (¿por qué?).

2.

$$\int_{-\infty}^{+\infty} x^{2n}e^{-x^2} dx = \Gamma(n + 1/2)$$

(recordemos que  $\Gamma(1/2) = \sqrt{\pi}$  y que  $\Gamma(n + 1) = n\Gamma(n)$ )

3.

$$\int_{-\infty}^{+\infty} \Phi_n^{(0)}\Phi_m^{(0)} dx = \delta_{n,m}$$

siendo  $\Phi_k^{(0)}$  los autoestados ortonormales del oscilador armónico (Eq. (1)).

### 3) Construcción de la matriz hamiltoniana

Resolveremos el problema del cálculo de autoestados resolviendo el problema de autovalores para una matriz  $H$  con elementos de matriz  $h_{mn}$  (Ec. (3)). Truncaremos el problema considerando una matriz  $H$  de cierto tamaño  $N$ , es decir, tal que  $0 \leq n, m \leq N$  ( $N \sim 10$  bastará para calcular el estado fundamental y los primeros niveles excitados). En lo que sigue, consideramos que la perturbación del hamiltoniano se puede escribir como

$$\mathcal{H}_1 = \lambda V(x), \tag{4}$$

donde  $\lambda$  ser un parámetro pequeño. Para construir la matriz Hamiltoniana se sugiere:

1. A partir de nuestro anterior programa **traper.m**, construir una rutina **traperh.m** cuya sintaxis de llamada sea:

```
>>traperh(eps, a, b, n, m)
```

y que calcule los elementos de matriz

$$\langle \Phi_n^{(0)} | V(x) | \Phi_m^{(0)} \rangle = \int_{-\infty}^{+\infty} V(x)\Phi_n^{(0)}(x)\Phi_m^{(0)}(x)dx$$

Como antes se dijo, unos valores prácticos de  $a$  y  $b$  son  $b = -a = 10$  y en cuanto a **eps**, nos conformaremos con  $\text{eps}=1.e-10$ .

2. La rutina **traperh.m** a su vez llamará a la función **funcih.m**, que se encargará de evaluar el integrando, es decir,  $V(x)\Phi_n^{(0)}(x)\Phi_m^{(0)}(x)$  para la función  $V(x)$  que consideremos.
3. Construiremos una función que llamaremos **hamilt.m**, cuya primera línea será:

```
function H=hamilt(NN,lambda)
```

donde **NN** es el tamaño que asignaremos a la matriz hamiltoniana troncada y **lambda** es, por supuesto, el parámetro  $\lambda$  en la ecuación (4). Recordemos que los elementos de matriz  $h_{nm}$  vienen dados por la ecuación (3).

Es importante señalar que la matriz hamiltoniana es simétrica, con lo que haríamos bien en no duplicar los cálculos. Con que, por ejemplo, calculemos sólo los elementos  $h_{mn}$  con  $n \leq m$  será suficiente pues  $h_{mn} = h_{nm}$ .

## 2.2. Cálculo de autovalores y autovectores.

Los autovalores de nuestra matriz hamiltoniana troncada (generados mediante **hamilt.m**) aproximarán los valores de la energía para el estado fundamental y los primeros excitados tomando un tamaño de la matriz hamiltoniana suficientemente grande. Si la perturbación es suficientemente pequeña podemos truncar para valores pequeños del tamaño.

Antes de proseguir, será conveniente comprobar que estamos calculando bien la matriz hamiltoniana. Para ello, se puede considerar como ejemplo una perturbación del tipo  $\mathcal{H}_1(x) = \lambda x$  (para lo cual tomaremos  $V(x) = x$  en **funcih.m**). Es sencillo comprobar que las energías de los estados perturbados son, exactamente:

$$E_n = 2n + 1 - \lambda^2/4$$

Mediante el comando **eig** de Matlab podemos comprobar si las correspondientes matrices hamiltonianas para este  $V(x)$  tiene como autovalores estas energías. Así, por ejemplo, para  $\lambda = 2$  los autovalores exactos del Hamiltoniano son: 0, 2, 4, .... Entonces, si por ejemplo, tecleamos:

```
>> format long e;h=hamilt(8,2);eig(h)
```

obtenemos como respuesta:

```
ans =
```

```
8.721020892110687e-007
2.000082480090117e+000
6.037996301517302e+000
4.002718908166360e+000
8.244334918759611e+000
1.418376083223668e+001
1.865973317625135e+001
1.087137251087652e+001
```

que se aproxima a lo que esperamos; no salen los valores exactos porque hemos troncado el problema a una matriz de tamaño 8. Cuanto más pequeño sea  $\lambda$  y más grande el tamaño de truncamiento, mejor será la aproximación. Por ejemplo, con la llamada

```
>> format long e;h=hamilt(15,2);eig(h)
```

obtendríamos los 6 primeros autovalores con 8 cifras exactas.

Una vez que obtengamos estos resultados, podemos pasar a la siguiente fase: calcular algunos autovalores y autovectores utilizando nuestros algoritmos.

### 2.2.1. Cálculo de autovalores por el método de la potencia inversa

Como sabemos, el método de la potencia sirve para obtener el autovalor más grande en módulo (y el correspondiente autovector) de una matriz diagonalizable. Recordemos el algoritmo de la potencia para calcular el mayor autovalor en módulo,  $\lambda$ , de una matriz  $A$ :

**Algoritmo: Método de la potencia**

Input:  $A, v_0, \epsilon$

Output:  $\lambda, v$

- (1)  $\Delta = \epsilon + 1, v_0 = v_0 / \|v_0\|; n=0;$
- (2) Repetir mientras  $\Delta > \epsilon$
- (4)  $n=n+1;$
- (5)  $z = Av; \lambda = v^T z; v = z / \|z\|;$
- (6) Si  $n > 1$  entonces  $\Delta = |1 - \lambda_i / \lambda|$
- (7)  $\lambda_i = \lambda$
- (8) Fin

Este algoritmo, dada una matriz cuadrada  $A$  y un vector de partida  $v_0$ , converge al autovalor  $\lambda$  mayor en módulo y a su correspondiente autovector  $v$ .

Para proseguir con la práctica, deberemos construir la función **potencia.m** que implemente el anterior algoritmo; su primera línea será:

```
function [lambda,autov]=potencia(A,v0,eps)
```

donde  $A$  es una matriz cuadrada,  $v_0$  el vector para comenzar la iteración y  $eps$  la precisión requerida (nos conformaremos con  $1.e-8$ ).

En nuestro caso, nos interesa obtener el estado fundamental y los primeros estados excitados del oscilador armónico perturbado, donde consideraremos como ilustración el caso  $\mathcal{H}_1(x) = \lambda x^4$  (con la correspondiente modificación en **funci.m**). Para obtener el estado fundamental, que es el correspondiente al menor autovalor, se puede utilizar iteración inversa que, como sabemos, es equivalente a aplicar el método de la potencia sobre la inversa de la matriz problema. Para simplificar la tarea, dejaremos que Matlab calcule por nosotros la inversa, si bien sería preferible aplicar iteración inversa sin necesidad de calcular la matriz inversa, tal como se describió en clase.

En resumen, para calcular la energía del estado fundamental teclearíamos para, por ejemplo,  $\lambda = 0,1$  (y  $V(x) = x^4$ ), lo siguiente:

```
H=hamilt(10,0.1);A=inv(H);
```

Con esto tendremos la matriz hamiltoniana  $H$  y su inversa  $A$ . La aplicación del método de la potencia sobre  $A$  con un vector  $v_0$  adecuado (¿cuál?), permite calcular el mayor autovalor de  $A$ ,  $\lambda$ ; la energía del estado fundamental será entonces  $1/\lambda$ . En concreto, si nuestro programa funciona correctamente deberíamos obtener:

```
format long e;h=hamilt(10,0.1);A=inv(h);v=eye(10);v0=v(:,1);1/potencia(A,v0,1.e-8)
```

```
ans =
```

```
1.065285701336658e+000
```

que es uno de los valores propios (el menor) que obtendríamos al hacer  $\mathbf{eig}(\mathbf{h})$  (coinciden las 10 primeras cifras) y corresponde a la ecuación (2) con  $\lambda = 0,1$ ,  $E_0(\lambda) \approx 1,0619$ .

Para obtener los primeros estados excitados, se puede aplicar el método de la potencia inversa con desplazamiento. Se trata de considerar el hamiltoniano desplazado,  $H(\rho) = H - \rho I$ , y utilizar el hecho de que si  $\lambda$  es autovalor de  $H$ ,  $\lambda - \rho$  lo es de  $H_d$  (y corresponden al mismo vector propio). Escogiendo convenientemente  $\rho$ , se pueden calcular los primeros estados excitados mediante el método de la potencia basado en la inversa de  $H(\rho)$ ; a su vez, será necesario escoger de forma conveniente el vector inicial  $v_0$  para cada uno de estos estados.

No se darán en este guión detalles sobre cómo escoger estos desplazamientos  $\rho$  ni los vectores iniciales pues es posible deducir valores adecuados tras un análisis sencillo del problema.

Como ejercicio, se sugiere obtener las energías del estado fundamental y los 2 primeros estados excitados para los valores  $\lambda = 0,1, 0,2, 0,3$  (una matriz de tamaño 10 es más que suficiente).

Otra actividad interesante puede ser dibujar la energía del estado fundamental en función de  $\lambda$ . Si comparamos en el intervalo  $\lambda \in [0, 0,5]$  veremos como la diferencia entre el valor calculado y la aproximación asintótica dada (Ec. (2)) empieza a ser apreciable para  $\lambda > 0,1$  (con tomar matrices de tamaño 5 y calcular para 10 valores de  $\lambda$  será más que suficiente -si somos más exigentes el cálculo puede ralentizarse-).

También se sugiere probar para otro tipo de perturbaciones, como por ejemplo,  $\mathcal{H}_1 = \lambda x^3$ ,  $\mathcal{H}_1 = \lambda \sin(x)$  o cualquier otro ejemplo; para eso sirve hacer los programas: tenemos el problema “resuelto” para los ejemplos “que queramos”, aunque hay potenciales  $V(x)$  para los que nuestro método de la potencia fallará. Por ejemplo, aquellos que hagan que la energía del estado fundamental se reduzca mucho haciéndose negativa pueden causar que el autovalor del estado fundamental no sea el más pequeño en módulo.

Se propone como ejercicio adicional (por si queda tiempo, cosa difícil) resolver el problema del cálculo de autovalores mediante iteración inversa si necesidad de calcular la inversa de la matriz hamiltoniana (o las matrices desplazadas). Esto supone, como se discutió en clase, resolver en cada paso de la iteración un sistema de ecuaciones. Se sugiere utilizar el método de Gauss con pivotaje parcial.

Finalmente, planteamos una serie de cuestiones. ¿Se puede encontrar factorización de Cholesky para la matriz Hamiltoniana para valores de  $\lambda$  suficientemente pequeños?. ¿Por qué?. En consecuencia, se pide describir que pasos deberíamos seguir para obtener la energía del estado fundamental perturbado utilizando la factorización de Cholesky (para resolver los sistemas de ecuaciones del algoritmo de iteración inversa). ¿Se puede aplicar el mismo esquema para calcular, mediante desplazamientos, las energías de los estados excitados?.

## Referencias

- [1] Galindo y Pascual, “Mecánica Cuántica”, tomo II.