Tercera práctica voluntaria — Técnicas Numéricas

Profesor Francisco R. Villatoro — 15 de Noviembre de 1999

ENUNCIADO DE LA TERCERA PRACTICA

El objetivo de esta práctica es estudiar algunos algoritmos iterativos de resolución numérica de sistemas lineales utilizando Matlab.

El problema de valores para la ecuación no lineal de Schrödinger unidimensional (ENLS), que se puede escribir en forma adimensional como

$$\mathrm{i}\,\frac{\partial u}{\partial t} + \frac{\partial^2 u}{\partial x^2} + q|u|^2 u = 0, \qquad x \in \mathbb{R}, \quad t \geq 0,$$

donde q es una constante positiva, t es el tiempo, x el espacio, $u(x,t) : \mathbb{R}^2 \to \mathbb{C}$ es una amplitud compleja e $\mathrm{i} = \sqrt{-1}$. Esta ecuación en derivadas parciales tiene solución analítica calculable mediante una generalización no lineal de la transformada de Fourier (la transformada espectral inversa). Las soluciones de esta ecuación están formadas por N solitones en un fondo de radiación que decrece con el tiempo. Un solitón es una onda solitaria $u(x,t) = f(x-ct)$ que se mueve con una velocidad c y que es robusto ante interacciones mutuas a pares, es decir, tras una colisión entre dos solitones, éstos recuperan su forma pero con un cambio de fase.

La expresión matemática para un solitón de la ENLS es

$$u(x,t) = \frac{A}{\sqrt{q}}\mathrm{sech}\left\{\frac{A}{\sqrt{2}}\left((x-x_0) - Vt\right)\right\}\exp\left\{\frac{\mathrm{i}}{2}\left(V(x-x_0) + \left(A^2 - \frac{V^2}{2}\right)t\right) + \mathrm{i}\,\phi_0\right\},$$

donde A es la amplitud del solitón, V es su velocidad, x_0 es la posición inicial de su máximo (en $t=0$) y ϕ_0 es su fase inicial.

La solución exacta con N solitones (obtenida por Zakharov and Shabat (1972)) se puede escribir como (Gordon (1983))

$$u(x,t) = \frac{1}{\sqrt{q}}\sum_{j=1}^{N} u_j(x,t),$$

donde u_j es la solución del siguiente sistema de ecuaciones lineales

$$\sum_{k=1}^{N} M_{jk} u_k = \sum_{k=1}^{N} \frac{\gamma_j^{-1} + \gamma_k^*}{\lambda_j + \lambda_k^*} u_k = 1, \qquad j = 1, 2, \ldots, N,$$

$$\gamma_j = \exp\left(\frac{\lambda_j}{\sqrt{2}}(x - x_{j0}) + \frac{\mathrm{i}}{2}\lambda_j^2 t + \mathrm{i}\,\phi_{j0}\right),$$

$$\lambda_j = A_j + \mathrm{i}\,\frac{V_j}{\sqrt{2}},$$

donde z^* es el complejo conjugado de z y los cuatro parámetros reales $\{A_j, V_j, x_{j0}, \phi_{j0}\}$ representan la amplitud, velocidad, posición inicial de la amplitud máxima y fase inicial, respectivamente, del j-ésimo solitón.

Edita el siguiente fichero `soliton.m`:

```
A = [1 2 3]'; V = [1 2 0]'; x0 = A*0; phi0 = A*0;
q = 1; I = sqrt(-1); N=length(A);

lambda = (A + I*V/sqrt(2)); Mlambda = lambda*ones(1,N);
denominador = ( Mlambda + Mlambda' );

X=-20:0.1:20; NX = length(X);   T=-20:20; NT = length(T);
M=zeros(N,N);   b=ones(N,1);   u=zeros(NX,NT);

for it=1:NT, t=T(it);
 for ix=1:NX, x=X(ix);

  gamma = (exp( lambda/sqrt(2).*(x-x0)+I/2*lambda.^2*t+ ...
                                     I*phi0) )*ones(1,N);

  M = ( 1./gamma + gamma' )./denominador;

  u(ix,it) = sum(M\b);

 end
end

u = 1/sqrt(q)*u;

figure(1);   mesh(X,T,abs(u)')
figure(2);   contour(X,T,abs(u'),30)
figure(3);   mesh(X,T,log(abs(u)'))
```

Ejecutalo en Matlab. Requiere unos 14 Mflops (megaflops). Si tu ordenador es ”lento”, utiliza `X=-20:20` (sólo 1.4 Mflops).

1. Explica el cálculo de `denominador`, `M` y `u`, comprobando en detalle que calculan la expresión correcta.

2. Cuando ejecutas el fichero aparecen muchos mensajes de tipo

   ```
   Warning: Matrix is close to singular or badly scaled.
            Results may be inaccurate. RCOND = 4.523788e-020
   ```

   Explica dónde está el problema y el porqué ocurre.

3. Escribe una función en Matlab que te calcule la matriz `M(x,t)` para n solitones con amplitudes $A_j = 1, \cdots, n$, velocidades $V_j = n, \cdots, 1$, posición y fase iniciales $x_{0j} = \phi_{0j} = 0$. Como parámetros para la función usa x, y, n. Escribe una gráfica con el número de operaciones flotantes (`flops`) para la resolución del sistema lineal $M(5,1)\, u = b$ mediante el método directo `u=M(5,1,n)\b`, en función del número de solitones `n` (que es el orden de la matriz). Ajusta (mínimo-cuadráticamente) dicha curva mediante un polinomio (usa `polyfit`). ¿Cuál es el menor grado que da un ajusta suficientemente bueno?

4. Escriba de forma matricial el método iterativo de Gauss-Jacobi e implementelo en una función de Matlab (`x=jacobi(A,b,TOL)`, donde `TOL` es la tolerancia de corte de la iteración, pongamos, en norma infinito, es decir, la última iteración k es la que cumple $\|x^{(k+1)} - x^{(k)}\|_\infty < TOL$). Recuerda que $A = L + D + U$ en Matlab es `L=tril(A,-1)`, `D=diag(diag(A))` y `U=triu(A,1)`. ¿Cumple este sistema las condiciones necesarias para que los métodos de Gauss-Jacobi y de Gauss-Seidel converjan?. Si no es así, ¿puede reescalar o equilibrar el sistema de ecuaciones para que sean convergentes?

5. Escribe una gráfica con el número de operaciones flotantes (`flops`) para la resolución del sistema lineal $M(5,1)\, u = b$ mediante el método iterativo de Jacobi con tolerancia de 10^{-5}. Ajusta (mínimo-cuadráticamente) dicha curva mediante un polinomio (usa `polyfit`). ¿Cuál es el menor grado que da un ajusta suficientemente bueno?. Escribe también una gráfica con el número de iteraciones hasta alcanzar la tolerancia

en función del valor de esta tolerancia. Escribe finalmente una gráfica con el número de iteraciones hasta alcanzar la tolerancia de 10^{-5} en función del parámetro `x=-20:20` (en $M(x,1)$).

6. Haz los dos apartados anteriores para el método de Gauss-Seidel y contesta las mismas cuestiones que en el apartado anterior.

7. Escriba de forma matricial el método de sobrerrelajación sucesiva (SOR) basado en Gauss-Seidel y escriba una función Matlab que lo aplique (como en apartados anteriores). Determine el rango de valores de w tales que el método SOR converge (dibuje gráficamente el radio espectral de la matriz de coeficientes y aplique la condición de convergencia) para el sistema de ecuaciones lineales de con matriz de coeficientes $M(5,1)$ en función del número de solitones. ¿Puede estudiar el caso general? ¿Puede obtener el w óptimo (convergencia más rápida)?

8. Para tres valores de w para los que haya convergencia, ¿cuántas iteraciones se requieren para que la diferencia entre dos iteraciones sucesivas en norma 1 sea menor que 10^8?

9. Si quisiera obtener un w óptimo, en sentido de que conduzca a la mayor velocidad posible de convergencia, ¿cómo lo haría? Razone su respuesta y si se siente capaz calcúlelo.

10. Se puede aplicar el método del gradiente conjugado al problema

$$M(x,t)\, u = b.$$

¿Por qué? Se puede aplicar dicho método al problema

$$M(x,t)^* M(x,t)\, u = b,$$

donde M^* es la traspuesta conjugada. ¿Converge dicho método para este sistema de ecuaciones? En su caso, escriba un código en Matlab con dicho método y aplíquelo a dicho problema. Cúantas iteraciones (m) son necesarias para conseguir convergencia con una tolerancia de 10^{-5}. Dibuja una gráfica con m en función de n para dicha tolerancia.