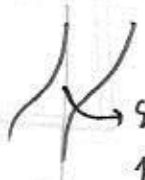
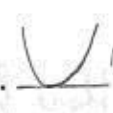


RESOLUCIÓN DE ECUACIONES NO LINEALES

$f(x) = g(x) \Leftrightarrow F(x) = 0$; con F no lineal.

• Habrá problemas con discontinuidades de $F(x)$ (p.ej. $\tan x$):  Si no tienes cuidado, aquí puede darte ∞

METODO DE LA BISECCIÓN

• Se basa en el Teorema de Bolzano: sirve para ceros con cambio de signo en su intervalo.  No 2.000

• Es un proceso iterativo recursivo:

• $\text{sgn}(a) \neq \text{sgn}(b) \Rightarrow$ Miro $\text{sgn}\left(\frac{a+b}{2}\right)$:

Miro si cambio de signo en $[a, p_1]$ o $[p_1, b]$ y repito

• Pasa cuando $f(p_i)$ esté a el cero de la precisión con la que estoy trabajando.

• La diferencia entre f de los dos extremos del intervalo es menor que la precisión con la que estoy trabajando. La raíz es cualquiera de los extremos, da igual con la precisión

• Si hay varios ceros, divide $[a, b]$ en subintervalos pequeños y aplico el algoritmo a aquellos en los que f cambia de signo.

METODO DE NEWTON-RAPSON

• Es más rápido y no hace falta un intervalo en el que se cambie de signo. Es en \bar{x} hay que conocer f' .

• Comenzamos con una aproximación \bar{x} :

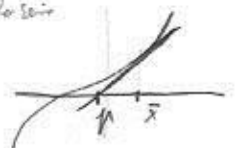
$$f(x) \approx f(\bar{x}) + f'(\bar{x}) \cdot (x - \bar{x})$$

Como existe una raíz auténtica p : $f(\bar{x}) + f'(\bar{x})(p - \bar{x}) = 0$

Así: puedo hallar una mejor aproximación si $f'(\bar{x}) \neq 0$

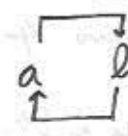
f' puede ser complejo, y tener derivada tipo de ignora

Nota que he anotado la raíz



$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}$$

PROBLEMAS

- Esto tiene problemas cerca de los puntos críticos (se ve tanto gráficamente como analíticamente), pues diverge.
- Hace falta una estimación inicial
- Cuidado con raíces patológicas a general (e.g. $\frac{1}{x}$)
- Para todos los problemas, inspección visual
- Puede darse la casualidad de entrar en un ciclo infinito: 

MÉTODO DE LA SECANTE

- Deriva numéricamente:

$$f'(p_{n-1}) = \frac{f(p_{n-1}) - f(p_{n-2})}{p_{n-1} - p_{n-2}} \rightarrow \text{Ejemplo secante}$$

$$p_n = p_{n-1} - \frac{f(p_{n-1})(p_{n-1} - p_{n-2})}{f(p_{n-1}) - f(p_{n-2})}$$

Requiere dos estimaciones iniciales (p_0 y p_1)

[Faint handwritten notes and a box at the bottom of the page]

INTERPOLACIÓN Y EXTRAPOLACIÓN

- Son técnicas enormemente útiles y peligoras.
- Interpolan es estimar el valor de una función en un punto dentro del rango a el cual conocemos los valores. (medida p.ej. = puntos de copiado de música)
- Extrapolan es estimarlo en una zona fuera de dicho rango. (tiempo que se va la batería que se le va.)
- El peligro radica en funciones de fuerte variación. (o medidas imprecisas o defectuosas)
- La utilidad está en la facilidad de derivar e integrar funciones interpolantes polinómicas, p.ej.
- ¿Cómo lo hacen?

Taylor

- Requiere muchas derivadas a cuanto queremos extrapolar a puntos lejanos.
- Así, en localidad nos entropen el método.

Polinomio

- Si tenemos n puntos, podemos trazar un único polinomio de grado $n-1$ que pase por todos. (Unicidad de n ecuaciones lineales con n incógnitas)
- Ahora, para n altos, los polinomios tienen fluctuaciones grandes e impredecibles; además de ser más costosas computacionalmente.
- Y más que el polinomio, lo que nos interesa es el valor del polinomio en ciertos puntos. Esto lleva a los polinomios de Lagrange: (solo de lectura a que el orden)

$$L_{n,k}(x) = \prod_{\substack{i=0 \\ i \neq k}}^n \frac{(x-x_i)}{(x_k-x_i)}$$
$$P_n(x) = \sum_{k=0}^n f(x_k) L_{n,k}(x)$$

$\{x_i\}$ = conjunto de puntos

→ Hemos a el ser visto en adelante que luego esto es función y nos ante lo primero itera de lagrange

$L_{n,k}$: Se hace 1 si $x = x_k$
Nosotros otros que divididos 0

Queda un sistema lineal de ecuaciones con $n+1$ incógnitas para b_i ($i \in \{0, \dots, n\}$)

Háctico

Usamos la subrutina `polint`: (interpolo en una palabra de loguex, de una intención \Rightarrow eficiente)

`polint(x, y, n, xinterp, yinterp, error)`

`x[1, ..., n]` \rightarrow contiene los datos

`y[1, ..., n]`

`yinterp` es el valor interpolado para el punto `xinterp`

El `polint` devuelve un vector
error nos dice cuánto varía `yinterp` si yo aumento el orden polinomial. NO se da el error intrínseco por una interpolación

Recordar: comience usar n bajos.

Para splines:

`call spline(x, y, n, y2, y21, y22)` \rightarrow se lo llama UNA SOLA VEZ

`y21` e `y22`, opcionales, son los valores de las derivadas en los extremos

`x[1, ..., n]` \rightarrow contiene los datos

`y[1, ..., n]`

`y2[1, ..., n]` es un vector de salida que recibe la siguiente subrutina: (llenar los derivados, segundo, para `spline` uso otro `split` y el lugar de `fl.` 3 era segundo derivado)

`call splint(x, y, y2, n, xinterp, yinterp)` \rightarrow se lo llama las veces que queramos

Como `polint`

Módulo:

`polint` \rightarrow `mf_interpoli`

`spline / splint` \rightarrow `mf_spline`

INTEGRACIÓN NUMÉRICA

· Evaluamos integrales definidas mediante la aproximación:

$$\int_a^b f(x) dx \approx \sum_{i=0}^n w_i f(x_i)$$

con w_i ciertos pesos.

· El empuje a x puede ser equiespaciado o no serlo. Esto último permite mejorar la precisión y/o la velocidad.

INTERPOLACIÓN POLINÓMICA

· La primera aproximación es usar polinomios de Lagrange: $f(x) = \sum_{k=0}^n L_{n,k}(x) f(x_k)$

$$\int_a^b f(x) dx = \int_a^b \underbrace{P_n(x)}_{\text{calculable}} dx + \text{error}$$

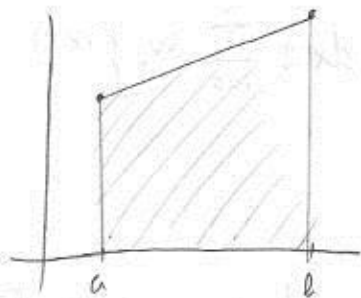
$A_i, w_i = \int_a^b L_{n,i}(x) dx$, que se pueden sacar previamente.

n	REGLA	ERROR
1	TRAPECIO	$(b-a)^3$
2	SIMPSON	$\left[\frac{(b-a)}{2}\right]^5$
3	SIMPSON (3/8)	$\left[\frac{(b-a)}{3}\right]^5$
4	BOOLE	$\left[\frac{(b-a)}{4}\right]^7$

\Rightarrow Si $b-a > 1$, el error va a crecer
 \Downarrow
Conviene usar intervalos
pequeños

REGLA DEL TRAPEZIO

$$\int_a^b f(x) dx = \frac{b-a}{2} [f(a) + f(b)] + O((b-a)^3)$$



REGLA DE SIMPSON

$$\int_a^b f(x) dx = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)] + O(h^5)$$

$$h = \frac{1}{2}(b-a)$$

$$x_0 = a$$

$$x_1 = x_0 + h$$

$$x_2 = x_1 + h$$

Zimatek

REGLA DE SIMPSON 3/8

$$\int_a^b f(x) dx = \frac{3}{8} h [f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)] + O(h^5)$$

REGLA DE BOOLE

El problema con más costo es la evaluación de f (los otros son sumas)

Notar, como siempre:

• Comienza MIRAR qué quieres obtener.

• lo ideal es usar:

• Intervalos pequeños

• Grados no muy altos

• Veamos qué ocurre dependiendo de la cantidad de intervalos:

• Aplicamos la regla del trapecio al cálculo de $\int_a^b f(x) dx$

• Dividimos $[a, b]$ en n intervalos $\Rightarrow h = \frac{b-a}{n}$

• Usar n grandes aumenta la cantidad de evaluaciones de f

• El error resulta ser un polinomio en potencia pura de h :

$$I_T = I + k_1 h^2 + k_2 h^4 + k_3 h^6 + \dots$$

• El error de Simpson, para evitar sobrecargas, n debe ser par. (los otros triplets también a h)

• Ahora bien, al aumentar n , aumenta la cantidad de operaciones y los errores se van acumulando.

• Nótese que la regla de Simpson es mejor que la del trapecio porque converge para n mayor ($n \sim 20$) y hay menos evaluaciones.

• Hay un método preciso que se aprovecha del error: calcula I para diferentes h y extrapola

para $h \rightarrow 0$. ^{No es una regla para evitar problemas de extrapolación} Se puede comprobar que cada vez que se hace así, se elimina un término del polinomio en h^2 para el error. Además, todo esto con muy pocas evaluaciones de la función.

• Intuitivamente, al hacer pocas evaluaciones, los resultados son erróneos pero **MUY PRECISOS**, así que la extrapolación funciona de maravilla.

BTW, podría hacerse sucesivas extrapolaciones lineales, que se puede demostrar equivalentes a un orden n

Esto define a el error que dentro de un nivel

CVADRATURA GAUSSIANA

- Hasta ahora hemos usado integración por interpolación polinómica mediante Lagrange.
- La integral será exacta cuando el integrando sea un polinomio.
- Ahora bien, podría elegirse un muestreo no equiespaciado. Con esto, tengo el doble de grados de libertad ($2n+2$)
↳ por tablas
- Voy a exigir que para un polinomio de orden $2n-1$ la integral sea exacta. Es decir, voy a hacer falta la mitad de puntos. (porque los algoritmos serán más eficientes)
- Esto requiere el uso de polinomios ortogonales (harán falta trabajar con sus ceros, derivadas...)
 - Los abscisas son sus ceros
 - Los pesos exigen calcular productos escalares
 - Es necesario un cambio de variable para ir del intervalo original a aquél en el que los polinomios están definidos.
- En Moodle hay una tabla con los tipos de función que según con qué polinomio dan resultados exactos.
- Nosotras utilizaremos polinomios de Legendre, que aciertan la integral si el integrando es un polinomio (como las fórmulas de Newton-Cotes cerradas). Si usamos orden n , dan polinomios de orden $\leq 2n-1$
↳ pta

INTEGRALES IMPROPIAS

ES FUNDAMENTAL REPRESENTAR SINGULARIDADES EN f

- Hay más fórmulas para integración numérica tras interpolación polinómica que no evalúan la función en los extremos. (fórmulas abiertas)
- Pero ello, utilizan sólo puntos interiores para calcular el polinomio interpolante:

$$x_0 = a + h$$

$$x_1 = a + 2h$$

\vdots

$$x_n = b - h$$

y se integra entre a y b .

$n=0$:

$$\int_a^b f(x) dx = 2h f(x_0) + O(h^3)$$

$n=1$:

$$\int_a^b f(x) dx = \frac{3h}{2} [f(x_0) + f(x_1)] + O(h^3)$$

$$n=2: \int_a^b f(x) dx = \frac{4h}{3} [2f(x_0) - f(x_1) + 2f(x_2)] + O(h^5)$$

- Como en los integrales usuales, muchos subintervalos + Runge funciona muy bien.
- De hecho, se pueden incluso usar fórmulas cerradas en los subintervalos, abiertos y abiertos en los extremos.

CAMBIOS DE VARIABLE

Cambios como:

$$t = \frac{1}{x} \quad (\text{útil si hay } \infty \text{ y el integrando va } \sim \frac{1}{x^2})$$

$$t = \sqrt{x-a} \quad (\text{si el integrando tiene } \sim \frac{1}{\sqrt{x-a}})$$

$$t = \sqrt{x+l} \quad (\text{si el integrando tiene } \sim \frac{1}{\sqrt{x+l}})$$

$$t = e^{-x} \quad (\text{si el integrando tiene exponencial y hay } \infty)$$

Son útiles.

Ala hora de usar varios métodos, hay que tener siempre ojo para ver qué método es bueno, e ir tanteando.

DIVISIÓN DEL INTERVALO

- Es una técnica **UTILÍSIMA**: aislar un pequeño intervalo con la singularidad, ^{es alto variación} en el que nos concentramos; y el resto, integrarlo con Riemann sin problemas.
- El aprovechamiento de la propiedad es también muy útil.

" ∞ "
Hay funciones que como caen muy rápido, ya valen el 0 de la máquina en $\log(x)$. Así, se puede hacer $\int_a^{\log(x)}$

PRÁCTICA

Hay un módulo en algunos métodos.

Tratamiento de integrales impropias

Consideraremos que una integral es impropia si se dan algunos de los siguientes supuestos:

- ✓ El integrando tiende a un valor finito en alguno de los límites del intervalo de integración, pero no puede evaluarse en dichos puntos.
- ✓ El límite superior del intervalo de integración es ∞ o el inferior $-\infty$.
- ✓ Tiene una singularidad integrable en alguno de los límites del intervalo de integración.
- ✓ Tiene una singularidad integrable en algún punto conocido dentro del intervalo de integración.

En ningún caso pueden tratarse mediante métodos numéricos integrales imposibles como $\int_1^{\infty} \frac{1}{x} dx$, $\int_{-\infty}^{\infty} \cos(x) dx$, ...

Como norma general para todo tipo de integrales es muy útil representar la función y dividir el intervalo de integración para aislar en lo posible los puntos "más complicados". En particular una integral impropia podrá integrarse con un algoritmo que no evalúe la función en los extremos (por ejemplo con fórmulas de Newton-Côtes abiertas) y/o con un cambio de variable adecuado. Algunos ejemplos útiles de cambios de variable son los siguientes:

$$\int_a^b f(x) dx = \int_{1/b}^{1/a} \frac{1}{t^2} f\left(\frac{1}{t}\right) dt \quad ab > 0$$

Apropiado si: $a > 0$ y $b \rightarrow \infty$ (o $a \rightarrow -\infty$ y $b < 0$) y la función decrece al menos como x^2 .

$$\int_a^b f(x) dx = \int_0^{\sqrt{b-a}} 2t f(a+t^2) dt \quad b > a$$

Apropiado si el integrando diverge como $(x-a)^{-1/2}$.

$$\int_a^b f(x) dx = \int_0^{\sqrt{b-a}} 2t f(b-t^2) dt \quad b > a$$

Apropiado si el integrando diverge como $(x-b)^{-1/2}$.

$$\int_a^{\infty} f(x) dx = \int_0^e f[-\log(t)] \frac{dt}{t}$$

Apropiado si el integrando decrece exponencialmente.

DERIVACIÓN NUMÉRICA

En algoritmos MUY inestable debido a la acumulación de errores.

Lo mismo es hacer $\frac{f(x+h) - f(x)}{h}$ para valores pequeños de h , pero sale un error.

El error de truncamiento es (Taylor), $O\left(\frac{f''(x_0)}{2} h^2\right)$, variable.

Hay un error de redondeo que, para h exacto, vale $\sim \frac{2 \Delta f(x_0)}{h} \sim \frac{\epsilon}{h}$ ↑ mínimo del ruido

Ahí, para h alto, esto se va al carajo.

Intuitivamente:

$x_0 = 10^{-3} \pm 10^{-9}$ ↑ por eso estoy esperando a brío
 $h = 0.0001$
 $\Rightarrow x_0 + h = 10^{-3} 001 \pm 10^{-6} \Rightarrow$ Estoy perdiendo precisión en h

Hay un valor óptimo de h que minimiza el error total, pero no sabemos cuál es.

En la práctica, salvo que las funciones sean bonitas (muchas evaluaciones) o solo conocidas derivadas

cientos puntos, hay que evitar la derivada numérica y usar derivación analítica.

En caso de que las necesitemos, hay fórmulas que derivan numéricamente mediante polinomio interpolante.

(exactamente igual que la integral numérica)

Las fórmulas de Neville son difíceles de pedirle cosas

$$x_0, x_0+h, x_0+2h$$

$$x_0, x_0+h, x_0+h$$

$$x_0, x_0-h, x_0-2h$$

Una evaluación más \Rightarrow la línea (además es lo que nos da de los intervalos de interpolación etc \Rightarrow no está bien)

Notar que la derivada del polinomio de Lagrange no tiene porque coincidir con la de la función. (a menos que sea un polinomio)

Además, para más inri, órdenes altos oscilan mucho \Rightarrow la derivada se va al carajo.

Se podría utilizar un método como el de Runge. Es de hecho, recomendable hacerlo para familiarizarse

con F . \rightarrow Intenta el error h pequeña por pérdida de precisión, pero la convergencia por que que te da.

Para segundas derivadas, lo mismo se trata en una interpolación polinómica. La actual de 4 puntos es precisa.

Una forma rápida de llegar a las fórmulas es desarrollar a serie:

$$f(x_0 + h) = f(x_0) + f'(x_0)h + \frac{1}{2} f''(x_0)h^2 + \dots$$

$$f(x_0 - h) = f(x_0) - f'(x_0)h + \frac{1}{2} f''(x_0)h^2 + \dots$$

Sumando y restando salen tanto las fórmulas como los errores.



ECUACIONES DIFERENCIALES ORDINARIAS

• Saber resolverlas numéricamente es vital a física. (intentar antes resolverlas analíticamente)

con C.I.,
• Trabajamos en 1º orden, pues los de orden superior equivalen a sistemas de 1º orden.
(te derivas de modo que se vuelva sus la otra variable)

• Dadas unas C.I., los métodos de resolución se hacen en desarrollos en serie (o al menos aquellas más inmediatas):

$$y(t_a + h) = y(t_a) + h y'(t_a) + \frac{1}{2} h^2 y''(t_a) + \dots$$

A 1º orden, ME TENGO QUE QUEDAR AQUÍ (la EDO = da y')
 $f(t_a, y(t_a))$

notar que para significar lo equispaciado la variable independiente.

MÉTODO DE EULER

$$y(t_{i+1}) = y(t_i) + h f(t_i, y_i) + O(h^2)$$

siendo la EDO $y' = f(t, y)$

• Considero (t_0) , esto procede iterativamente.

• Gráficamente, estoy avanzando considerando las tangentes (líneas rectas).

• El problema es que esto acumula errores a cada iteración.

• Para sistemas ^{propiedades s-o}, se procede con vectores: $\vec{y}(t_{i+1}) = \vec{y}(t_i) + h \vec{f}(t_i, \vec{y}_i) + O(h^2)$

donde la dimensión de \vec{y} es el n.º de físicas.

MÉTODOS DE ORDEN SUPERIOR

Mi pregunta es: ¿puedo ir más allá del desarrollo a 1º orden? $\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f$

Bases $\{\alpha_1, \alpha_2, \beta_1\}$ y $\alpha_1 f(x+\alpha_1, y+\beta_1) = f(x, y) + \frac{h}{2} \frac{d}{dt} f(x, y(t))$; donde $f = \frac{dy}{dt}$

desarrollen serie: $\alpha_1 f(x+\alpha_1, y+\beta_1) \approx \alpha_1 \left[f(x, y) + \frac{\partial f}{\partial x} \Big|_x \alpha_1 + \frac{\partial f}{\partial y} \Big|_y \beta_1 \right]$

$$\cdot \alpha_1 = 1$$

$$\cdot \alpha_2 = \frac{h}{2}$$

$$\cdot \beta_1 = \frac{h}{2} f(x, y)$$

• Así, a luego de derivar f puedo, para la siguiente, evaluar f en $(x + \frac{h}{2}, y + \frac{h}{2} f(x, y))$

• Este es el método de Runge-Kutta de orden 2:

$$w_0 = y(0)$$

$$w_{i+1} = w_i + h f \left[x_i + \frac{h}{2}, w_i + \frac{h}{2} f(x_i, w_i) \right]$$

• Constantemente, utilizar la derivada no en un extremo de su intervalo, sino el punto intermedio.

(es a dividir esto al 2 Euler)

INCISO

• Utilizar siempre módulos naturales ^{ya lo usamos} para que el h venga acorde con el problema.

FIN INCISO

• Esto puede, de hecho, llevarse hasta tercer orden ($O(h^3)$): realiza una iteración de Euler, dos de Runge-Kutta de orden 2, y una extrapolación mediante el método de Euler; pero, al final, hacer una media ponderada.
Usa el método de Euler en el punto medio del intervalo

Finalmente como no conoces la solución real no puedes comprobar si el paso elegido es correcto. Así, hay que girarse por:

- Consistencia con resultados límite conocidos
- Consistencia para diferentes personas

C.C.

• El método de diferencias finitas (se verá en una hoja de problemas) es muy poderoso, aunque ahora vamos otro: el método del disparo.

• Nosotros, a priori, propagamos información a partir de C.I. ¿Cómo trabajamos con C.C.?

• Como sólo habrá una curva que cumpla ambas C.C., podemos desarrollar un método de tanteos: introducción de C.I.:

$$y(0) = y_0$$

$$y'(0) = \alpha; \text{ con } \alpha \text{ un número cualquiera}$$

• Si con α se cumple la otra C.C., buen acierto. Si no, hay que probar otro α hasta que se cumpla.

• Este tanteo se puede mejorar: tengo para cada α una función que en $x = x_1$ debe valer y_1 :

$$y(x_1, \alpha) - y_1 = 0$$

estoy buscando raíces de función!!!!

• Se puede aplicar el método de Newton: $\alpha_i = \alpha_{i-1} - \frac{f(x_1, \alpha)}{f'(x_1, \alpha)}$, que necesita

la derivada respecto a α !!!

• Esto se solventa resolviendo otra EDO. Se verá en problemas.

~~Para problemas de Sturm-Kovalevskii, el método del cociente se aplica al valor propio, siendo la derivada el signo arbitrario. (para las soluciones: depende de las ... etc. multiplicativas)~~

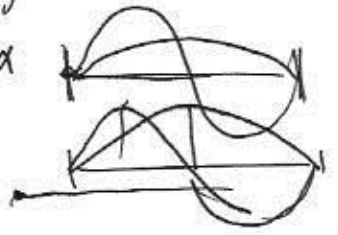
• Si hay varios parámetros, habrá que usar el método de Newton en varias variables, que es un poco.



C.I. para z

$y(x, \alpha)$ es el valor x de la solución de la EDO cuyas C.I. son $y(a) = y_a$

$y'(a) = \alpha$



Como $y(a, \alpha) = y_a \quad \forall \alpha \Rightarrow \frac{\partial y(a, \alpha)}{\partial \alpha} = z(a, \alpha) = 0$

$\frac{\partial z(a, \alpha)}{\partial x} = \frac{\partial}{\partial \alpha} \left. \frac{\partial y(x, \alpha)}{\partial x} \right|_{x=a} = \frac{\partial}{\partial \alpha} \underbrace{y'(a, \alpha)}_{\alpha \forall \alpha} = \frac{\partial \alpha}{\partial \alpha} = 1$

$\frac{d^2 y(x, t)}{dx^2} = -\alpha y(x, t)$
 $y' = 1$

$w_i^{(k)}$ es el valor de w_i en la k -ésima iteración. $\Delta_i^{(k)}$ es la diferencia entre iteraciones. Como los extremos son fijos, sus valores varían $\Rightarrow \Delta_0^{(k)} = \Delta_{N+1}^{(k)} = 0 \quad \forall k$

$f(x, y, y') = p'(x)y' + q'(x)y + r(x)$

$\frac{w_{i+1}^{(k)} - 2w_i^{(k)} + w_{i-1}^{(k)}}{h^2} + \frac{\Delta_{i+1}^{(k)} - 2\Delta_i^{(k)} + \Delta_{i-1}^{(k)}}{h^2} = \dots$

$= f(x_i) \frac{\Delta_{i+1}^{(k)} - \Delta_{i-1}^{(k)}}{2h} + q(x_i) \Delta_i^{(k)} + r(x_i)$

En diferencias finitas, cada espere $w_{i+1} = f(w_i, w_{i-1})$; w_{i-1} tiene información acerca de las derivadas: se puede calcular con fórmulas a 3 puntos.

Notar que en diferencias finitas, y'' da siempre lugar a una matriz ^{tridiagonal} simétrica. Si f es función de y' ; van a aparecer valores propios complejos (resonante).

Si el sistema es ideal, la aproximación es PERFECTA, y no hace falta repetir el proceso de iteración:

$$\frac{w_{i+1} - 2w_i + w_{i-1}}{h^2} = p_i \frac{w_{i+1} - w_{i-1}}{2h} + q_i w_i + r_i$$

$$p_i = p(x_0 + ih)$$

$$q_i = q(x_0 + ih)$$

$$r_i = r(x_0 + ih)$$

~~$$\frac{1}{h^2} + p_i$$~~

$$\frac{1+p_i}{h^2} w_{i-1} - \left(\frac{2}{h^2} + q_i \right) w_i + \left(\frac{1-p_i}{h^2} \right) w_{i+1} = r_i$$

$$\frac{\partial^2 y}{\partial x^2} - \frac{1}{c^2} \frac{\partial^2 y}{\partial t^2} = 0$$

2) Modos normales: $y(x,t) = \mathbb{I}(x) T(t)$: $\mathbb{I}'' T - \frac{1}{c^2} \mathbb{I} \ddot{T} = 0$

$$\frac{\mathbb{I}''}{\mathbb{I}} = \frac{1}{c^2} \frac{\ddot{T}}{T} = -k^2$$

$$T = A \cos kt + B \sin kt$$

$$\frac{\mathbb{I}''}{\mathbb{I}} = -k^2 \Rightarrow \mathbb{I}'' = -k^2 \mathbb{I} \text{ constante, } f(x,y,t) = 0 \cdot y' - k^2 \cdot y + 0 \quad \begin{matrix} p=0 \\ q=-k^2 \\ r=0 \end{matrix}$$

$$\frac{1}{h^2} w_{i-1} - \left(\frac{2}{h^2} - k^2 \right) w_i + \frac{1}{h^2} w_{i+1} = 0$$

$$\frac{1}{h^2} w_{i-1} - \frac{2}{h^2} w_i + \frac{1}{h^2} w_{i+1} = -k^2 w_i \Rightarrow \text{Ecuación de valores propios!!!}$$

Extremos $w_0=0$
 $w_N=0$

Variables: $w_{N+1} = w_{N-1}$ (periodicidad)

Para el método de los dígitos, no de parámetro de dígitos la propia constante K (aunque K sea arbitraria) $y = A \cos Kx + B \sin Kx$ $\Rightarrow y' = -K \sin Kx + K \cos Kx \Rightarrow y'(0) = K$

$$y'' = f = -K^2 \cdot y$$

$$\frac{\partial f}{\partial y} = -K^2$$

$$\frac{\partial f}{\partial y'} = 0$$

$$\frac{\partial f}{\partial K^2} = -y$$

Valor en $C_1 \cdot \cos(x)$

~~$$y(a) = u_1(a) = 0.0$$~~

~~$$y'(a) = u_2(a) = 1.0$$~~

~~$$\frac{\partial y(a, \alpha)}{\partial \alpha} = u_3(a) = 0.0$$~~

$$\frac{\partial Z}{\partial x} = \frac{\partial}{\partial \alpha} \left. \frac{\partial y(x, \alpha)}{\partial x} \right|_{x=a} = \frac{\partial}{\partial \alpha} \frac{y'(a, \alpha)}{1 \cdot \alpha} = 0$$

Zimatek

SISTEMAS DE ECUACIONES LINEALES

- Aparecen con mucha frecuencia en física.
- Queremos programas de álgebra matricial. Hay una amplia literatura sobre el tema.
- Hay incluso programas específicos para cada tipo de matriz.
- Nosotros vamos a ver algunos más generales.

METODO DE GAUSS-JORDAN

• Esperamos por el método de Gauss: mediante transformaciones elementales, convertimos un sistema a diagonal.

- Dividimos la 1ª fila por a_{11}
- Multiplicamos la 1ª fila por a_{i1} y se la restamos a la i -ésima fila

$$\left. \begin{array}{l} \cdot \text{Dividimos la 1ª fila por } a_{11} \\ \cdot \text{Multiplicamos la 1ª fila por } a_{i1} \text{ y se la restamos a la } i\text{-ésima fila} \end{array} \right\} \begin{pmatrix} 1 & \dots & \dots \\ 0 & \dots & \dots \\ 0 & \dots & \dots \\ 0 & \dots & \dots \end{pmatrix}$$

- Dividimos la 2ª fila por a_{22}
- Multiplicamos la 2ª fila por a_{i2} y se la restamos a la i -ésima fila $\forall i$

$$\left. \begin{array}{l} \cdot \text{Dividimos la 2ª fila por } a_{22} \\ \cdot \text{Multiplicamos la 2ª fila por } a_{i2} \text{ y se la restamos a la } i\text{-ésima fila } \forall i \end{array} \right\} \begin{pmatrix} 1 & 0 & \dots \\ 0 & 1 & \dots \\ \vdots & \vdots & \ddots \\ 0 & 0 & \dots \end{pmatrix}$$

$$\begin{pmatrix} \dots & \dots \\ 1 & 0 \\ \dots & \dots \\ 0 & 1 \\ \dots & \dots \end{pmatrix}$$

haciendo esto tl. a la columna de términos independientes, hallamos las soluciones.

- Esto es problemático: con involucra divisiones, si algún término diagonal es 0, lo hemos liado
 - Para solucionarlo, voy colocando los elementos más grandes de la matriz en la diagonal permitiendo filas y columnas. (pivotar)
- Lo voy haciendo así: \Rightarrow a cada columna poner el elemento mayor de la diagonal (o lo más cercano a él)

MÉTODO DE ELIMINACIÓN GAUSSIANA - Método F

La idea es transformar la matriz en una triangular superior.

$$E'_j = E_j - a_{ji} \frac{E_i}{a_{ii}} \quad \forall i$$

Demora, hay que pivotar. Aquí es por cada pivote solo con los datos por debajo de la diagonal, pues buscamos una matriz triangular.

Truco de programación:

• Anidar la matriz de coeficientes en columnas con los términos independientes.

(para seguir con los intercambios de filas)

• Pivotar: \cdot Se usa i que fila está el término mayor. Si $a_{ii} = 0$, salir.

• Intercambiar fila (i) y fila (j) para intercambiar filas

• No aplicar a las filas con a lugar de $a(i, j)$; a fila $(i), j)$

↓
Al intercambiar solo etiquetar, no intercambiar todos los términos

DESCOMPOSICIÓN LU

La idea es descomponer la matriz de coeficientes en una triangular superior y otra triangular inferior. L

$$Ax = b \quad (A = LU)$$

$$LUX = b \Rightarrow Ly = b \xrightarrow{\text{resuelto directo}} y$$

$$UX = y \xrightarrow{y = Ux} x$$

↓
resuelto directo

La ventaja es que, al ser L y U triangulares, la resolución del sistema es inmediata.

• El planteamiento directo de $LU = A$ lleva a un sistema no lineal de n^2 ecuaciones con $n^2 + n$ incógnitas (los elementos de L y U).

- Como nos salvan en incógnitas, ignoramos, por comodidad, que la diagonal de L sea 1.
- Este algoritmo tiene una ventaja inmensa: desarrollado los determinantes por rows, veas que el determinante de una matriz triangular es el producto de los términos de la diagonal principal. Como la diagonal de L es 1, el dte. es el producto de los elementos diagonales de U .
- Hay un algoritmo muy eficiente (algoritmo de Crout) que, eligiendo adecuadamente el orden de resolución de las ecuaciones, resuelve $A=LU$ de manera muy eficiente. (El pivote para no dividir entre 0 \Rightarrow ojo que si pivota, el dte. cambia de signo)
- A^{-1} se saca ignorando $AA^{-1} = I$ y resuelto nos da un sistema cuyo término independiente es cada una de las columnas de la matriz I .
- En la práctica, buscamos una submatriz que demuestre:
 - Los elementos extradiagonales de L encijados a los ceros de U
 - Un vector con los pivotañetes
 - Un elemento con el signo del determinante (por los pivotañetes)

DIAGONALIZACIÓN

- En lugar de resolver $|A - \lambda I| = 0$, ^{es muy aburrido (vale la pena atentar a los lineales)} se busca una matriz Z tal que $Z^{-1}AZ$ sea diagonal. Se puede incluso buscar una triangular: los valores propios están en la diagonal.
- Vamos a aprovechar de teorías bellas y útiles: una matriz hermitica es \mathbb{R} -diagonalizable mediante, además, una base ortogonal.
 - \rightarrow es una matriz ortogonal y \mathbb{R}/\mathbb{C} son matrices triangulares superiores / inferiores, respectivamente.
- Hay una serie de algoritmos iterativos (QR y QL) implementados. Son pesados, $O(N^3)$ por iteración.
 - \downarrow Más estable respecto a los errores de redondeo
 - \rightarrow triangular superior con una subdiagonal por debajo de la principal
- Si la matriz es tridiagonal, es $O(N)$; y si es de Hessenberg, $O(N^2)$.

• Hay otros métodos (de Householder) que conviene:

Matriz real \rightarrow matriz de Hessenberg

Matriz simétrica \rightarrow matriz tridiagonal

• Si las matrices son, a general, hermitiana, con el teo:

$$(A + iB)(v_1 + iv_2) = \lambda(v_1 + iv_2)$$

\Downarrow

$$(Av_1 - Bv_2) + i(Av_2 + Bv_1) = \lambda(v_1 + iv_2)$$

\Downarrow

$$\left(\begin{array}{c|c} A & -B \\ \hline B & A \end{array} \right) \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \lambda \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \text{ y estas en coma}$$

\hookrightarrow Matrices simétricas

$$\lambda(v_1) = \begin{pmatrix} A \end{pmatrix} \begin{pmatrix} v_1 \end{pmatrix} - B \begin{pmatrix} v_2 \end{pmatrix} = \begin{pmatrix} A & -B \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \text{ (fila por columna segunda)}$$

$$\lambda(v_2) = B \begin{pmatrix} v_1 \end{pmatrix} + A \begin{pmatrix} v_2 \end{pmatrix} = \begin{pmatrix} B & A \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \text{ (otra)}$$

hacer como el otro, da

EDP

- Cada una es de su padre y de su madre: nosotros, aplicamos, eso sí, siempre diferencias finitas.
- hablenos sin símbolos son más complicados pero requiere expresiones de L, C .

ECUACIÓN DE DIFUSIÓN

$$\frac{\partial p}{\partial t} = \kappa \nabla^2 p$$

• Normas métodos:

→ tanto por datos como por \mathbb{R} no es repetible

• Estable: los errores no crecen con el tiempo \Rightarrow Convergente

• Consistente: al tomar $\Delta t \rightarrow 0$, se llega a la derivada \rightarrow los algoritmos más simples traen más error!!!

• Discretización: (2D)

$$x_i = i h$$

$$t_n = n \tau$$

1^{er} derivada \Rightarrow fíjate a 2 puntos (por estabilidad)

2^{er} derivada \Rightarrow fíjate a 3 puntos

$$\frac{T_{i,n+1} - T_{i,n}}{\tau} = \kappa \frac{T_{i+1,n} - 2T_{i,n} + T_{i-1,n}}{h^2}$$

$$T_{i,n+1} = T_{i,n} + \frac{\kappa \tau}{h^2} [T_{i+1,n} - 2T_{i,n} + T_{i-1,n}]$$

FTCS
Forward
Time
Central
Space

Ani, conocida la distribución inicial, esto es todo

o, naturalmente, $T_n = A^n T_0 \Rightarrow$ el error vale $E_n = A^n E_0$. Tomando normas:

$$\|E_n\| = \|A^n\| \cdot \|E_0\|$$

o, debe tener $\|A^n\| \leq 1$ para que la error no se demore

ahora bien, la teoría nos dice que $\|A^n\| \geq |\mu_{\max}^n| = |\mu_{\max}|^n \Rightarrow$ si el máximo valor propio es ≤ 1 ,

el algoritmo es estable. Si no, la hees liado. El otro valor propio TODO (EXAMEN). Busca por μ_{\min} (los otros o cuando diré)

Aquí, la condición de estabilidad es $\frac{\Delta t}{h^2} < 1/2 \Rightarrow$ El Δt no puede ser el tiempo que tarda la temperatura a difundirse de un elemento a otro.

- Con las diferencias son pequeñas, los algoritmos van a ser lentos.
los derivados espaciales se calculan en $n+1$
- Una alternativa es BTCS: evaluar los derivados hacia atrás e insertar la correspondiente matriz.
 La ventaja es que todas las valores propios son menores que 1 \Rightarrow sigue estable. Computacionalmente es más costoso.
- El método lineal (de Crank-Nicolson) combina ambos algoritmos: (los métodos ya se veían lo que viene)

$$-r T_{i-1, n+1} + 2(1+r) T_{i, n+1} - r T_{i+1, n+1} = r T_{i-1, n} + 2(1-r) T_{i, n} + r T_{i+1, n}$$

$$r = \frac{\Delta t}{h^2}$$

\downarrow
 Hay que insertar matriz y multiplicar por ella
 (o un vector de ceros si tienes independientes)

A pesar de que tiene inestabilidades, usándose los que se veían

ECUACIÓN DE LAPLACE

- Es el límite cuando $t \rightarrow \infty$ de la ecuación de difusión.
- Así, se resuelve la ecuación de difusión a el límite de estabilidad, llegando así a un método iterativo:

$$M_{i, j, n+1} = \frac{1}{4} (M_{i+1, j, n} + M_{i-1, j, n} + M_{i, j+1, n} + M_{i, j-1, n})$$

(antes parecíamos a los puntos de alrededor)

el método de Gauss-Seidel aprovecha valores ya calculados para acelerar la convergencia:

$$M_{i, j, n+1} = \frac{1}{4} (M_{i+1, j, n} + M_{i-1, j, n+1} + M_{i, j+1, n} + M_{i, j-1, n+1})$$

ECUACIÓN DE ONDAS

• Ecuación en 1D. Discretizando:

$$u_{i,n+1} = \left(\frac{c\tau}{h}\right)^2 [u_{i+1,n} - 2u_{i,n} + u_{i-1,n}] + 2u_{i,n} - u_{i,n-1} \quad V_n \neq 0$$

!!!
↓
Revisado es la derivada

$$u_{i,1} = \left(\frac{c\tau}{h}\right)^2 \frac{u_{i+1,0} - 2u_{i,0} + u_{i-1,0}}{2} + u_{i,0} + \tau u_{i,0}$$

• El problema es el diseño en métodos explícitos: hay un límite de estabilidad (el tiempo para la evolución de difusión)

ECUACIÓN DE SCHRÖDINGER

• Normalmente se suelen utilizar unidades naturales:

$$m = \frac{1}{2} \quad L = 1 \Rightarrow \text{el tiempo viene fijado}$$

$$\hbar = 1$$

• A primera vista, salvo por $V(x)$, tengo una ecuación de difusión con el detallito de la i . Eso nos obliga a trabajar con números complejos (línea explícita, línea separada parte real e imaginaria).

• Discretizando:

$$\Psi_{j,n+1} = \Psi_{j,n} - i\tau \left(\frac{-\Psi_{j+1,n} + 2\Psi_{j,n} - \Psi_{j-1,n}}{h^2} + V_j \Psi_{j,n} \right) \quad \text{FTCS ligera}$$

• Esto tiene un límite de estabilidad. Para amplificar, hacemos Crank-Nicolson. Si definimos

$$U_{j,k} \equiv - \frac{\delta_{j,k+1} - 2\delta_{j,k} + \delta_{j,k-1}}{h^2} + V_j \delta_{j,k}$$

$$\text{FTCS y } \Psi_{n+1} = [1 - i\tau H] \Psi_n$$

BTCS ser:

$$\Psi_{n+1} = \Psi_n - i \tau H \Psi_{n+1}$$

(Ejercicios: equiborlo)
de una

Estable
No que quic
hudo

Ans: C-Na:

$$\Psi_{n+1} = \Psi_n - i \frac{\tau}{2} H [\Psi_n + \Psi_{n+1}]$$

⇓

$$\Psi_{n+1} = \left[1 + \frac{i\tau}{2} H \right]^{-1} \left[1 - \frac{i\tau}{2} H \right] \Psi_n$$

Estable
heiso
Algo pudo

Ejercicios de una Admido

por lo que, sin más que invertir ^{que similitud no vez y etona en} matrices, obtenemos un método iterativo estable.

Si no queremos invertir, resulta que el algoritmo equivale a

$$\Psi_{n+1} = (G^{-1} - 1) \Psi_n ; \text{ con } G = \frac{1}{2} \left[1 + \frac{i\tau}{2} H \right]$$

$$\text{Si llenas } X = G^{-1} \Psi \Leftrightarrow G X = \Psi \quad (\text{resolvamos entre de una vez triangular})$$

⇓

$$\underline{\Psi_{n+1} = X - \Psi_n}$$

Ejercicio 11:

$$\Psi_{n+1} = \left[1 + \frac{i\tau}{2} H \right]^{-1} \left[2 \mathbb{1} - \left[1 + \frac{i\tau}{2} H \right] \right] \Psi_n = \underbrace{\left[2 \left[1 + \frac{i\tau}{2} H \right]^{-1} - \mathbb{1} \right]}_{G^{-1}} \Psi_n$$

MÉTODOS DE MONTE CARLO

- Es quizás la técnica de cálculo más poderosa.
- Surge para hacer frente al problema del tiempo de cálculo en integrales N -dimensionales.

Ejemplos ideas:

- Quiero $\int_a^b f(x) dx = I$

- Tengo una distribución de probabilidad $g(x)$

- Hago $\int_a^b g(x) \frac{f(x)}{g(x)} dx = \left\langle \frac{f(x)}{g(x)} \right\rangle \Rightarrow$ Me transformo integrales
en valores medios!!!

- El valor medio se puede estimar como $\left\langle \frac{f}{g} \right\rangle = \sum_i \frac{f}{g}(z_i)$; con z_i
unos puntos que siguen la distribución de probabilidad g .

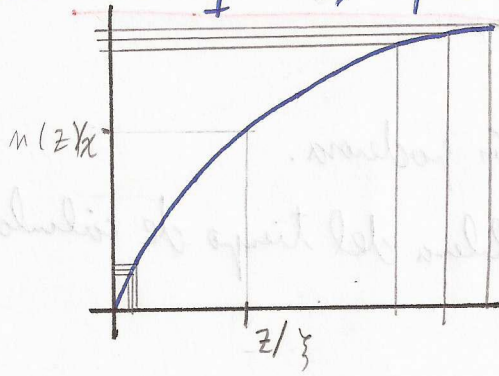
Aún, el mismo problema es muestrear una distribución de probabilidad no uniforme (los ordenadores emplean distribuciones uniformes).

Algunos ejemplos típicos son el problema de Buffon o el cálculo del área de un círculo mediante generación de números aleatorios.

Un ordenador es capaz de generar números "aleatorios" uniformemente distribuidos. El problema
en 32 bits, tiene un periodo de 2^{32}

es que a física se le aparecen distribuciones no uniformes.

Para generar una distribución cualquiera, mapeamos $[0, 1]$ a nuestro intervalo:



\Rightarrow Va a haber más puntos en la parte izquierda!!!

Se puede demostrar que la curva que mapea a la distribución de probabilidad es la función de probabilidad acumulada:

$$n(z) = \int_0^z p(x) dx$$

Se puede demostrar que a medida que avanza el tiempo, con el tiempo que el tiempo sea más...

¡ Para calcular integrales reciento resolver otras!

$$x = \int_0^{\xi} p(x) dx \Rightarrow x = P(\xi) - P(0)$$

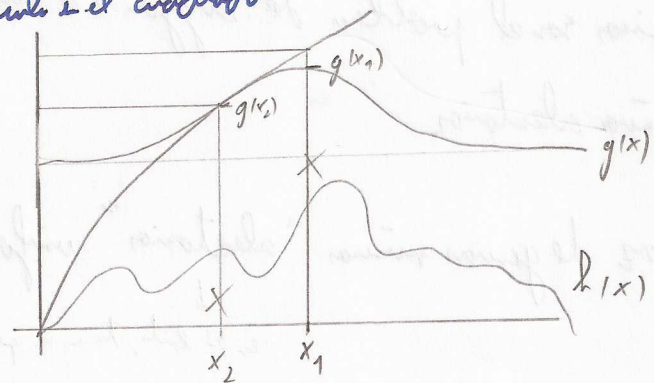
$$\Downarrow$$

$$\xi = F(x) \rightarrow \text{reciento esto}$$

Zmatek

El truco es el siguiente: (método de eliminación) (Método de Metropolis, no variante, es útil)

- Se aplica una distribución conocida $g(x)$ que siempre sea mayor que la distribución $h(x)$ a analizar.
- Se usa el truco del círculo y el cuadrado



- Hacemos n aleatorios entre $g(x)$
- Si es por debajo de $h(x)$, se queda
- Cuando n aleatorios entre $g(x)$
- Si es por debajo de $h(x)$, se queda

(es como generar más al azar y ver qué los que caen bajo la curva: voy a obtener (por definición) una distribución de probabilidad)