

1 Ondas de sonido

Una perturbación de pequeña amplitud en un fluido va a crear zonas de compresión y rarefacción que alternan en el tiempo y espacio en la dirección de propagación (el movimiento oscilatorio de las partículas se propaga como una onda y son ondas longitudinales: las partículas se mueven en la misma dirección que la dirección de propagación)

Condiciones iniciales Consideramos un fluido ideal con las variables velocidad, densidad y presión (dependen de x y t) en el estado de equilibrio $v_0 = 0, \rho_0, p_0$ (p_0, ρ_0 constantes en el caso homogéneo, variando lentamente en espacio y/o tiempo en el caso inhomogéneo) y hacemos unos cambios en estas variables añadiendo unas perturbaciones: v, ρ', p' que son muy pequeñas de forma que se pueden omitir los términos de segundo orden (o mayores) en las ecuaciones de los fluidos. Definimos

$$c_s = \sqrt{\gamma \frac{p_0}{\rho_0}}$$

$$c_s = \left(\frac{\partial p}{\partial \rho} \right)_s$$

Las variables de las ecuaciones de los fluidos devienen:

v

$$\rho = \rho_0 + \rho'$$

$$p = p_0 + p'$$

$$\text{con } |v| \ll c_s, |p'| \ll p_0, |\rho'| \ll \rho_0$$

Evolución Queremos estudiar la evolución en el tiempo de las perturbaciones: p', ρ', v Las ecuaciones de los fluidos después de linearizar devienen:

$$\text{continuidad: } \frac{\partial \rho'}{\partial t} + \rho_0 \nabla \cdot v = 0$$

$$\text{movimiento: } \frac{\partial v}{\partial t} + \frac{1}{\rho_0} \nabla p' = 0$$

$$\text{energía: (El proceso es adiabático) } p' = c_s^2 \rho'$$

Notaciones

N número de dimensiones, $(x_i)_i$ vector $\in \mathbb{R}^N$, $f(x) = (f_i(x))_i$ vector $\in \mathbb{R}^N$ (como la velocidad), $g(x)$ scalar $\in \mathbb{R}$ como la densidad, $p(x) = p(x)_{i,j}$ matriz $\in \mathbb{R}^N \times \mathbb{R}^N$ como f_c de abajo

$$\nabla \cdot f(x) = \text{div}(f) = \sum_i \frac{\partial f_i}{\partial x_i}$$

$$\nabla \cdot g(x) = \text{div}(g) = \sum_i \frac{\partial g}{\partial x_i}$$

$$\nabla g(x) = \frac{\partial g}{\partial x_i}$$

$$\nabla \cdot p(x) = \sum_j \frac{\partial p_{i,j}}{\partial x_j}$$

$$x \cdot y = \sum_i x_i y_i \text{ para } x_i, y_i \text{ vectores}$$

De forma numérica las ecuaciones de los fluidos de arriba se expresan:

en el momento inicial definimos:

$$u_m = \rho$$

$$u_c = \rho v$$

$$u_e = \frac{p}{\gamma - 1} + \frac{1}{2} \rho |v|^2$$

En cada paso de tiempo

1. se calculan los flujos definidos:

$$f_m = \rho v$$

$$f_c = (\rho v_i v_j)_{i,j} + p I_N$$

matriz de dimension (N,N) N = numero de dimensiones del espacio y I_N la matriz identidad

$$f_e = u_e p v$$

2. y se recalculan u_m, u_e, u_c (aqui se pueden usar esquemas diferentes que se configuran en constants.py: lax-fr("lf") o first generation("fg")). Tambien en este paso se aplican las condiciones de contorno el tipo se configura en soundwave.boundary_conditions.py y se han implementado "repeat" y "refl" (en el caso se la esquema "fg" estas se pueden aplicar también en el paso intermedio dependiendo del parámetro bcStep de constants.py ("interm" o "final"))

$$\frac{du_m}{dt} = -\nabla \cdot f_m$$

$$\frac{du_e}{dt} = -\nabla \cdot f_e$$

$$\frac{du_c}{dt} = -\nabla \cdot f_c$$

3. y las variables v, p, ρ :

$$v = \frac{1}{\rho} u_c$$

$$p = (\gamma - 1)(u_e - \frac{1}{2\rho}|u_c|^2)$$

$$\rho = u_m$$

dt depende de los valores de las variables en el momento que se calcula ρ, p, v , resolución (nint), y el tipo de esquema numérico usado (depende del parametro fcfl que es diferente para las esquema lax-fr y first generation (que se implementaron en la práctica))

de forma analítica:

Después de hacer cálculos (y suponiendo el caso general inhomogeneo en tiempo y espacio : $\rho_0 = \rho_0(x, t), p_0 = p_0(x, t) \implies c_s = c_s(x, t)$ con p_0, ρ_0, c_s monótonas, variando bastante lentamente en el espacio y tiempo de tal forma que se pueden omitir terminos de segundo orden o mas en multiplicaciones de derivadas temporales o espaciales de estas y las perturbaciones) las perturbaciones de las variables verifican las ecuaciones:

$$\frac{\partial}{\partial t} \left(\frac{1}{c_s^2(x, t)} \frac{\partial p'}{\partial t} \right) = \nabla^2 p'$$

$$\frac{\partial}{\partial t} \left(\frac{1}{c_s^2(x, t)} \frac{\partial \rho'}{\partial t} \right) = \nabla^2 \rho'$$

$$\frac{\partial}{\partial t} \left(\frac{1}{c_s^2(x, t)} \frac{\partial \Phi}{\partial t} \right) = \nabla^2 \Phi \text{ donde definimos } v = \nabla \Phi \text{ considerando el fluido irrotacional}$$

Medio homogéneo

Si la densidad y presión de equilibrio son constantes en tiempo y espacio ($\rho_0, p_0 \text{ constantes} \implies c_s \text{ const}$) :

$$\frac{1}{c_s^2} \frac{\partial^2 p'}{\partial t^2} = \nabla^2 p'$$

$$\frac{1}{c_s^2} \frac{\partial^2 \rho'}{\partial t^2} = \nabla^2 \rho'$$

$$\frac{1}{c_s^2} \frac{\partial^2 \Phi}{\partial t^2} = \nabla^2 \Phi$$

En 2d (o 3d) no hay una solución general como en el caso 1d (cuando cualquier solución se podía escribir de forma $F(x + c_s t) + G(x - c_s t)$ o - por la T. Fourier - como una superposición de ondas harmónicas viajan a la derecha o izquierda)

Pero $F(k \cdot x - \omega t)$ donde $\omega^2 = c_s^2 k^2$ es una solución (el cartón)

Ondas esféricas, cilíndricas que no son planas

Medio inhomogéneo independiente de tiempo $p_0 = p_0(x), \rho_0 = \rho_0(x) \implies c_s = c_s(x)$ monótonas, variando lentamente..

$$\frac{1}{c_s^2(x)} \frac{\partial^2 p'}{\partial t^2} = \nabla^2 p'$$

$$\frac{1}{c_s^2(x)} \frac{\partial^2 \rho'}{\partial t^2} = \nabla^2 \rho'$$

$$\frac{1}{c_s^2(x)} \frac{\partial^2 \Phi}{\partial t^2} = \nabla^2 \Phi$$

Análogo a la solución del caso homogéneo de la onda plana: $p(x, t) = a \cos(\phi(x))$ donde $\phi(x, t) = k \cdot x - \omega t$ con la amplitud a const y ω, k constantes verificando la relación de dispersión $\omega^2 = c_s^2 k^2$ intentamos buscar soluciones de forma $a(x, t)e^{i\phi(x, t)}$ (aproximación WKB): donde definimos

$$\omega(x, t) = -\frac{\partial \phi}{\partial t}$$

$$k(x, t) = -\nabla \phi$$

Resolver la ecuación genérica

$$\frac{1}{c_s^2(x)} \frac{\partial^2 p}{\partial t^2} = \nabla^2 p$$

Reemplazando la solución WKB approx. $p(x, t) = a(x, t)e^{i\phi(x, t)}$ en la ecuación y con las definiciones de ω y k de arriba después de hacer los cálculos y asumiendo que las variaciones en la amplitud son muy pequeñas de forma que podemos omitir términos de segundo orden en las derivadas espaciales y temporales de a llegamos a:

- $\omega^2(x, t) = c_s^2(x)k^2(x, t)$ (la relación de dispersión es válida de forma local)
- y la ecuación de la evolución de la amplitud: $\frac{\partial a}{\partial t} + c_g \cdot \nabla a = -\frac{1}{2} \frac{a}{|k|c_s} \left(\frac{\partial \omega}{\partial t} + c_s^2 \nabla \cdot k \right)$

con $c_g = \frac{\partial \omega}{\partial k} = c_s \frac{k}{|k|}$ (k es un vector)

de la relación de dispersión \implies

$$\frac{\partial \omega}{\partial t} + c_g \cdot \nabla \omega = 0$$

$$\frac{\partial k}{\partial t} + c_g \cdot \nabla k = -k \cdot \nabla c_g$$

$$\frac{\partial \phi}{\partial t} + c_g \cdot \nabla \phi = 0$$

Usamos directamente la ecuación de conservación de energía para determinar la amplitud y no la de arriba (TODO):

$$\frac{\partial E}{\partial t} + c_g \cdot \nabla E = -E \nabla \cdot c_g$$

Al largo de un rayo $x_p(t)$ solución de :

$$\frac{dx}{dt} = c_g$$

$$x(0) = x_p$$

las dependencias de x se transforman en dependencias de t reemplazando x por $x_p(t)$

y reemplazando en las ecuaciones de arriba obtenemos las ecuaciones diferenciales:

$$\frac{d\omega}{dt} = 0$$

$$\frac{d\phi}{dt} = 0$$

$$\frac{dk}{dt} = -k \cdot \nabla c_g$$

$$\frac{da}{dt} = -\frac{1}{2} \frac{a}{|k|c_s} \left(\frac{\partial \omega}{\partial t} + c_s^2 \nabla \cdot k \right)$$

$$\frac{dE}{dt} = -E \nabla \cdot c_g$$

2 D

Las características: $\frac{dx_i}{dt} = c_s \frac{k_i}{\sqrt{k_0^2 + k_1^2}}, i \in 0, 1$

Consideramos el caso inhomogeneo solo en una dirección: (considerando la dirección e_1 , pero se puede generalizar fácil para una dirección arbitraria, $c_s(x) = c_s(x_1)$)

$$\frac{dk_0}{dt} = 0$$

$$\frac{dk_1}{dt} = -\frac{\partial c_s}{\partial x_1} |k|$$

al largo del rayo Las ecuaciones ahora no tienen mas derivadas parciales (solo dependen de t) y se pueden integrar más fácil.

De la primera se obtiene:

$$\omega(x_p(t), t) = constant$$

Haciendo unos cálculos:

$$\frac{dc_s}{dt} = \frac{\partial c_s}{\partial x} \frac{\partial x}{\partial t} = \frac{\partial c_s}{\partial x} c_s \implies \frac{dk}{dt} = -k \frac{1}{c_s} \frac{dc_s}{dt} \implies$$

$$\frac{d \ln(k)}{dt} = -\frac{d \ln(c_s)}{dt} \implies$$

$$k(x_p(t), t) c_s(x_p(t)) = constant$$

de forma similar $E(x_p(t), t) c_s(x_p(t)) = constant$

Relación entre las amplitudes de las variables p, ρ, v

Suponiendo que estas soluciones existen:

$$p' = P e^{i\phi}$$

$$\rho' = R e^{i\phi}$$

$$v = V e^{i\phi}$$

P, R, V complejos

de la relación de adiabaticidad: $|P(x, t)| = c_s^2(x) |R(x, t)|$

de la ecuación de movimiento:

$$\rho_0 \frac{\partial v}{\partial t} = -\nabla p' \implies -i \rho_0 \omega V e^{i\phi} = i k P e^{i\phi}$$

despues de simplificar, multiplicar cada lado con su conjugado(hay que expresar las amplitudes locales con el módulo porque pueden ser complejas):

$$|V|^2 = \frac{|P|^2}{\rho_0^2 c_s^2} \implies$$

$$|V| = \frac{1}{c_s \rho_0} |P| = \frac{c_s}{p_0 \gamma} |P|$$

es independiente de si el medio es homogéneo o no

es independiente del número de dimensiones

$$\text{De las relaciones entre las amplitudes} \implies \frac{\rho_0 |V|^2}{2} = \frac{|P|^2}{2 \rho_0 c_s^2}$$

$$E = \frac{|P|^2}{\rho_0 c_s^2} \text{ (la suma de las 2)}$$

$$\implies |P(x_p(t), t)|^2 \rho_0^{-1}(x_p(t)) c_s^{-1}(x_p(t)) = constant$$

Suponemos que podemos escribir para los puntos al largo del rayo $x_p(t)$:

$$|P| = c_s^k \rho_0^l p_0^m C_{pP}$$

$$|R| = c_s^k \rho_0^{l_1} p_0^{m_1} C_{pR}$$

$$|V| = c_s^k \rho_0^{l_2} p_0^{m_2} C_{pV}$$

con C_{pP}, C_{pR} y C_{pV} constantes (que dependen del punto x_p)

de la ecuación de conservación de energía al largo del rayo $x_p(t)$: $|P(x_p(t), t)| = \text{const} * c_s^{\frac{1}{2}}(x_p(t)) \rho_0^{\frac{1}{2}}(x_p(t)) \implies$

existen las relaciones: $k = 2l - \frac{1}{2} = \frac{1}{2} - 2m$ y $2m + 2l = 1$

k se puede dejar como parámetro ya que $\gamma = \text{const} = c_s^2 \rho_0^{-1} p_0^{-1}$ y multiplicando las ecuaciones con γ^p el parámetro $k \rightarrow k + 2p$

elegimos $k = l = \frac{1}{2}$, $m = 0$

de las relaciones entre las amplitudes de arriba

$$\frac{|P|}{|R|} = c_s^2 = \rho_0^{l-l_1} p_0^{m-m_1} \frac{C_P}{C_R}$$

Escribimos $c_s^2 = \gamma p_0 \rho_0^{-1}$ y la relación de arriba se tiene que cumplir $\forall x$ (porque consideramos c_s, ρ_0 y p_0 dependiendo de x)

$$\implies m - m_1 = 0, l - l_1 = 0 \text{ y } \frac{C_P}{C_R} = \gamma$$

$$\frac{|P|}{|V|} = c_s \rho_0 = \gamma^{-\frac{1}{2}} p_0^{\frac{1}{2}} \rho_0^{\frac{1}{2}} = \rho_0^{l-l_2} p_0^{m-m_2} \frac{C_P}{C_V}$$

$$\implies \frac{C_P}{C_V} = \gamma^{-\frac{1}{2}}, l - l_2 = \frac{1}{2}, m - m_2 = \frac{1}{2}$$

$$\implies m_1 = -1, l_1 = \frac{3}{2}, l_2 = 0, m_2 = -\frac{1}{2}$$

En conclusión: al largo del rayo $x_p(t)$

$$|P| = c_s^{\frac{1}{2}} \rho_0^{\frac{1}{2}} C_P$$

$$|R| = c_s^{\frac{1}{2}} \rho_0^{\frac{3}{2}} p_0^{-1} \gamma^{-1} C_P$$

$$|V| = c_s^{\frac{1}{2}} p_0^{-\frac{1}{2}} \gamma^{\frac{1}{2}} C_P$$

con C_P constante que depende de x_p ($C_P = |P(x_p, 0)| c_s^{-\frac{1}{2}} \rho_0^{-\frac{1}{2}}$ donde $|P(x_p, 0)|$ es el valor de la amplitud de la perturbación de la presión en el momento $t = 0$ y el punto x_p)

2 Práctica

Definimos una amplitud A(en soundwave_perturbation_params.py) muy pequeña y una función periodica h en el intervalo $[x_0, x_f]$ (estos están en constants.py) con amplitud máxima 1 y creamos las perturbaciones para que cumplan las relaciones entre las amplitudes de arriba:

$$p'(x, 0) = \gamma p_0 h(x)$$

$$\rho'(x, 0) = \rho_0 h(x)$$

$$v(x, t) = c_s h(x)$$

h periodica $\implies h$ se puede escribir como una superposición de ondas sinusoidales (componentes Fourier) que viajan a la derecha

Se pueden también definir superposiciones de funciones periodicas que pueden viajar en los 2 sentidos para generar las perturbaciones (condiciones iniciales) (definiendo A , cs Sign como se explicó arriba y el tipo de función para cada una : en `soundwave_perturbation_params.py`)

Wave packet Elegimos `functionType = "wave_packet"` in `soundwave_perturbation_params.py` (gauss wave packet: los parámetros z_0, z_c, W definidos en `sound_wave_packet_params.py`)

$$h(x) = e^{-\frac{(x-x_c)^2}{W^2}} \cos(2\pi k_0 \frac{x-x_0}{x_f-x_0})$$

2.1 Medio homogéneo

Evolución en el tiempo de las variables

$$A = 3e - 4$$

Como las 3 variables: p', ρ', v cumplen las mismas ecuaciones diferenciales consideramos $h_2(x, t)$ solución de la ecuación

genérica: $\frac{1}{c_s^2} \frac{\partial^2 h_2}{\partial t^2} = \frac{\partial^2 h_2}{\partial x^2}$ con c_s constante y condición inicial $h_2(x, 0) = h(x)$ y luego la solución de las 3 variables es multiplicar esta con los valores correspondientes para que cumplan las relaciones entre las amplitudes de arriba

La solución analítica es $h_2(x, t) = h(x - c_s t)$

el gráfico de la función inicial (en este caso el paquete de ondas gaussian) no cambia de forma solo se traslada a la derecha con la velocidad c_s

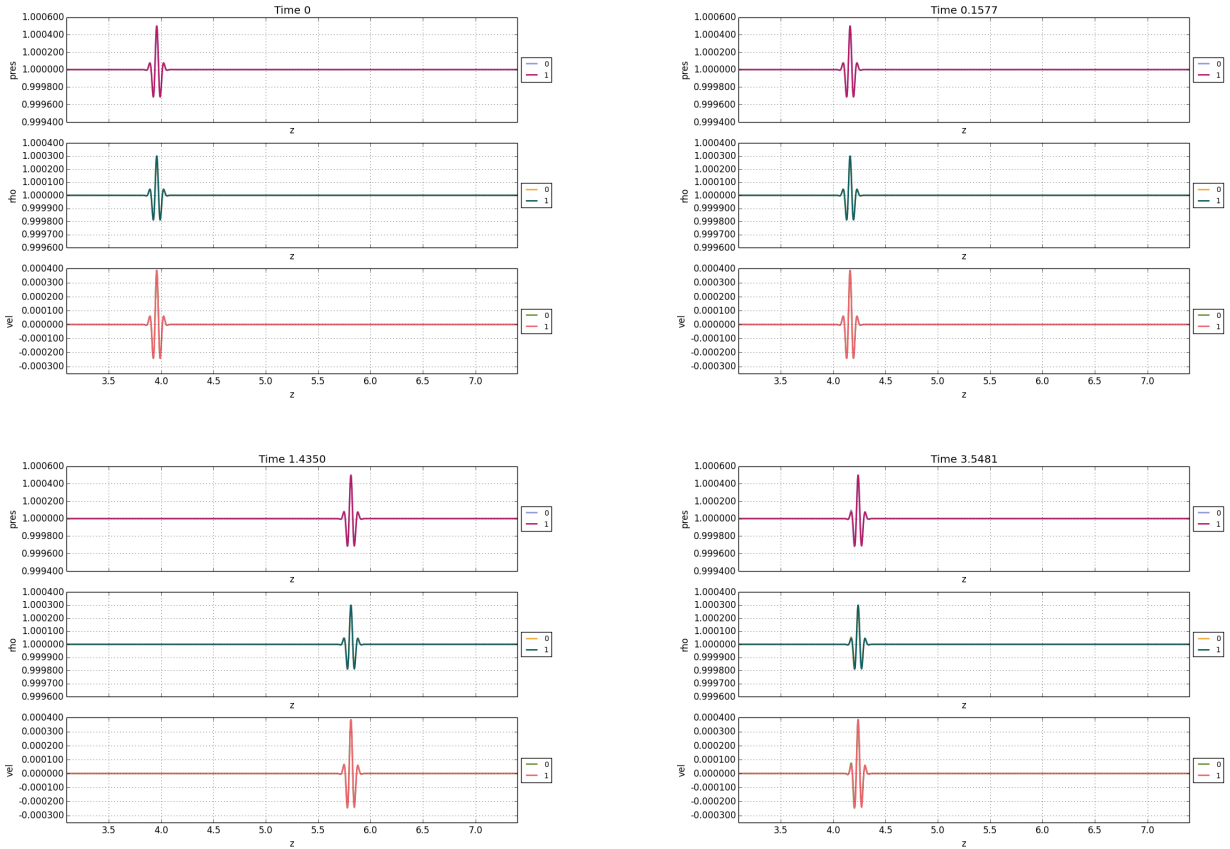


Figure 1: *El paquete de ondas en medio homogéneo en varios instantes del tiempo: boundary conditions: repeat*

si la resolución no es buena la amplitud calculada de forma numérica puede crecer. En este caso he tomado $n_{int} = 2048$ (definido en `constants.py`) y parece que la solución analítica dibujada por encima de la numérica no se desvía mucho de ella.

Fourier Equivalencias de las definiciones:

La transformada Fourier de la función h se define (omitiendo dominios): $F_h(k) = \int_{-\infty}^{\infty} h(x)e^{-ikx} dx$ para $h(x)$ definida arriba

calculo con mathematica (de forma analítica) la siguiente integral $f_1(k) = F_h(\frac{2\pi k}{x_f - x_0}) = \int_{-\infty}^{\infty} h(x)e^{-\frac{2\pi i k x}{x_f - x_0}} dx$

calculo con mathematica (de forma analítica) la transformada fourier de h con FourierTransform usando FourierParameters $0, -2\pi$ y obtengo $f_2(k)$

Compruebo de forma numerica y analítica que $f_1(k) = f_2(\frac{k}{x_f - x_0})$

calculo con mathematica (de forma analítica) la transformada fourier de h con FourierTransform sin especificar FourierParameters que por defecto son $0, 1$ y obtengo $f_3(k)$. De la documentación de mathematica y también se puede comprobar de forma analítica y numérica $f_3(k) = \sqrt{\frac{1}{2\pi}} \int_{-\infty}^{\infty} h(x)e^{ikx} = \frac{1}{\sqrt{2\pi}} f_1(\frac{(x_f - x_0)k}{-2\pi})$

De la teoria sé que si defino los coeficientes: $c(k) = \frac{1}{x_f - x_0} f_1(k)$

puedo escribir $f(x) = \sum_{k=-\infty}^{\infty} c_k e^{\frac{2\pi i k (x - x_0)}{x_f - x_0}}$

observo que en python los coeficientes calculados (de forma numérica) con scipy.fft son $cp_n = \frac{1}{x_f - x_0} f_1(k_n)$ donde $k_n = \frac{n}{x_f - x_0}, n = -N/2..N/2$ (el array se obtiene con `fftfreq(N,dz)`) así que

quiero representar de forma analítica y numérica $f_1(k) = c(k)(x_f - x_0)$

de forma analítica: calculo de forma analítica en mathematica $f_1(k)$ (calculando la integral de forma explícita o obtengo $f_2(k)$ con FourierTransform y calculo $f_2(\frac{k}{x_f - x_0})$), luego calculo los valores $f_1(k_n)$

de forma numérica: calculo en python con scipy.fft cp_n , luego los valores $cp_n(x_f - x_0)$

los represento frente a $n (= k_n(x_f - x_0))$

quiero representar de forma analítica y numérica $f_3(k)$

$f_3(-2\pi k_n) = \frac{1}{\sqrt{2\pi}} f_1(n) \implies$ el gráfico es equivalente al gráfico de $\frac{1}{\sqrt{2\pi}}(x_f - x_0)cp_n$ frente a n

de forma analítica: calculo de forma analítica en mathematica $f_3(k)$ (calculando la integral de forma explícita o con FourierTransform y calculo, luego calculo los valores $f_3(-2\pi k_n)$)

de forma numérica: calculo en python con scipy.fft cp_n , luego los valores $cp_n \frac{x_f - x_0}{\sqrt{2\pi}}$

los represento frente a $(2\pi k_n)$

Resultado analítico:

después de hacer unos cálculos llego a

$$f_1(k) = \frac{W\sqrt{\pi}}{2}(g_1(k) + g_2(k)) \text{ con}$$
$$g_1(k) = e^{-\frac{\pi[\pi W^2(k+k_0)^2 + 2ix_c(x_f - x_0)(k+k_0) - 2k_0ix_0(x_f - x_0)]}{(x_f - x_0)^2}}$$
$$g_2(k) = e^{-\frac{\pi[\pi W^2(k-k_0)^2 + 2ix_c(x_f - x_0)(k-k_0) + 2k_0ix_0(x_f - x_0)]}{(x_f - x_0)^2}}$$

para calcular $f_3(k)$ reemplazo en la expresión de $f_1(k)$ k por $\frac{(x_f - x_0)k}{-2\pi}$ y k_0 por $\frac{k_0(x_f - x_0)}{2\pi}$ la multiplico con $\frac{1}{\sqrt{2\pi}}$ y despues de hacer unos cálculos:

$$f_3(k) = \frac{W}{2\sqrt{2}} e^{-ikx_c} (u_1(k) + u_2(k))$$

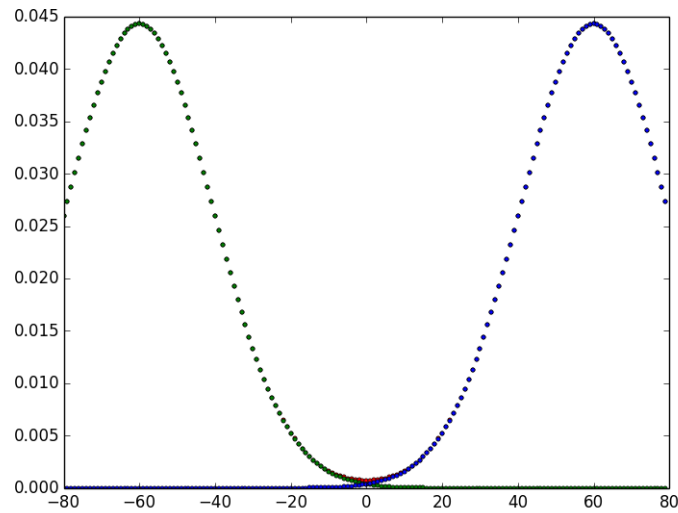
$$u_1(k) = e^{-\frac{W^2}{4}(k+k_0)^2} e^{-ik_0(x_0-x_c)}$$

$$u_2(k) = e^{-\frac{W^2}{4}(k-k_0)^2} e^{ik_0(x_0-x_c)}$$

Considerando $x_c = 0$ las exponenciales g_1 y g_2 son gaussianas con $w = \frac{x_f - x_0}{\pi W}$ la primera centrada en $-k_0$ y la segunda en k_0

si calculamos el módulo de cada una el modulo las constantes $abs(exp(-2k_0ix_0(x_f - x_0))) = abs(exp(2k_0ix_0(x_f - x_0))) = 1$ (corresponden al desplazamiento de fase por la elección de x_c (el centro de la función gauss) y x_0 (el centro de la función cos))

la amplitud máxima de cada una es $\frac{W\sqrt{\pi}}{2}$ igual que se ve en el gráfico por estar bien separadas (con valores : $z_0=3.100$, $z_f = 7.400$, $k_0 = 60$, $z_c = 3.745$, $W = 0.050$) : con rojo había hecho el plot de la función entera y con verde y azul de las 2 partes



Evolución de $f_1(k)$ en el tiempo $c_k * (x_f - x_0)$

Si escribimos $h_2(x, 0) = h(x) = \sum_{k=-\infty}^{\infty} c_k e^{2\pi i k \frac{x-x_0}{x_f-x_0}}$

$h_2(x, t) = h(x - c_s t) = \sum_{k=-\infty}^{\infty} c_k e^{2\pi i k \frac{x-c_s t-x_0}{x_f-x_0}}$

$h_2(x, t) = \sum_{k=-\infty}^{\infty} c'_k e^{2\pi i k \frac{x-x_0}{x_f-x_0}}$

c_s constante $\implies c_k = c'_k$

Igual que se ve en el gráfico $f_1(k)$ casi no varía en el tiempo (se representó también la solución analítica para hacer la comparación). La pequeña variación se debe a la resolución ($n_{int} = 2048$) (la solución numérica decrece un poco)

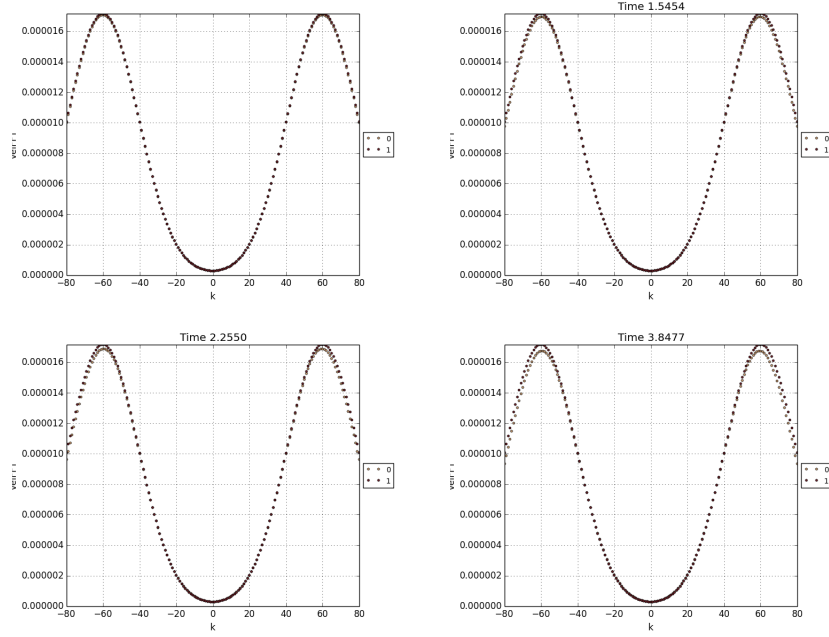


Figure 2: *evolución de $f1(k)$*

Condiciones de contorno: reflexión

cambiando el parametro periodicType in soundwave_boundary_conditions.py podemos elegir las condiciones de contorno ("refl" o "repeat")

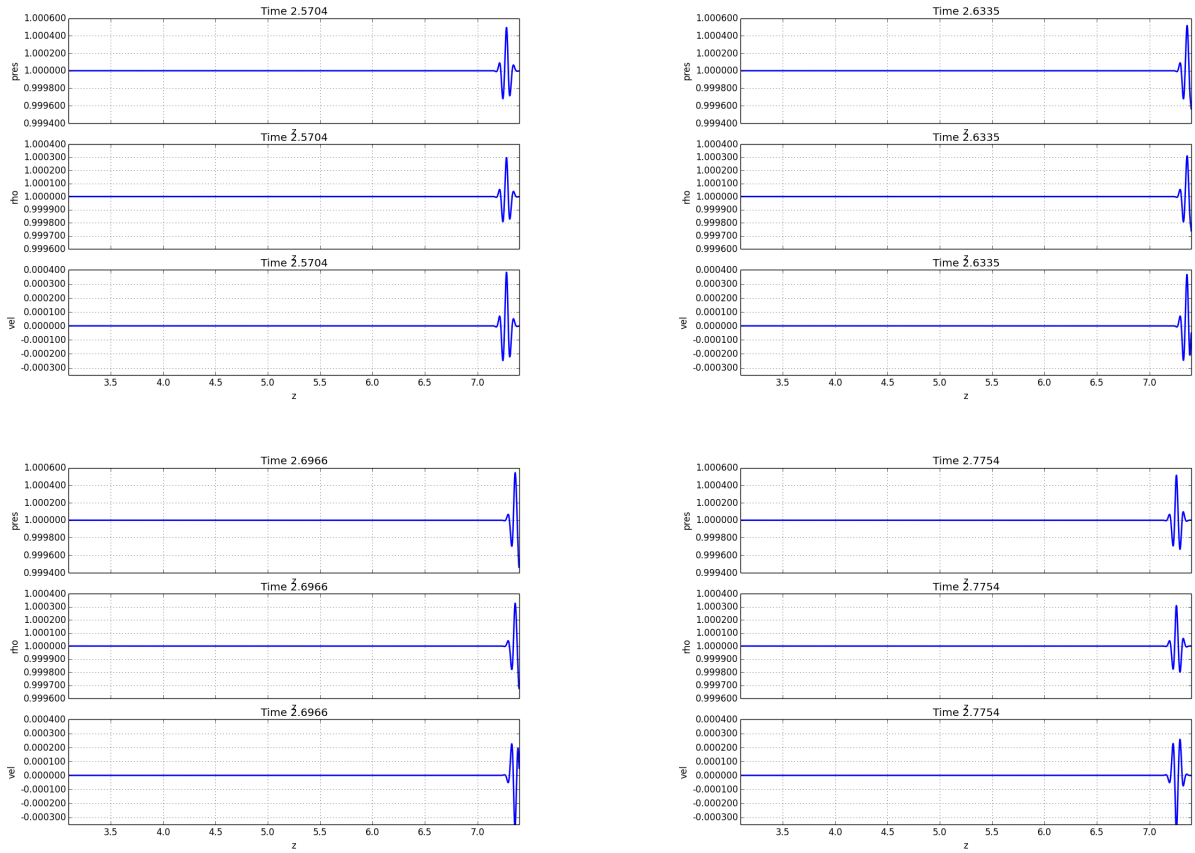


Figure 3: *Evolución en el tiempo de las variables boundary conditions: refl*

TODO: la solución analítica con periodicType="refl" no está todavía implementada

2.2 Medio inhomogéneo

p_0 const y $\rho_0(x)$ definida :

$$\rho_{00} + \frac{\rho_{01} - \rho_{00}}{2} \left[1 + \tanh\left(\frac{z - z_e}{w_e}\right) \right]$$

(las configuraciones del medio estan en soundwave_medium_params.py mediumType = "homog" o "inhomog" , ρ_{00} y los demás parámetros del caso inhomogéneo: z_e, w_e, ρ_{01})

ρ_0 **decrece** ($\rho_{00} > \rho_{01}$)

Evolución de la presión, densidad, velocidad, $\rho Curve = \frac{\rho - \rho_0}{\rho_0}$)

Las amplitudes de las perturbaciones de presión, densidad y velocidad no son mas constantes al largo de los rayos como en el caso homogéneo igual que se ve en el gráfico de arriba, la amplitud de la presión y $\rho Curve$ decrece y la de la velocidad crece (las time derivative no son mas 0 como en el caso homogéneo)

En el caso homogéneo la solución para $\frac{1}{c_s^2} \frac{\partial^2 h_2}{\partial t^2} = \frac{\partial^2 h_2}{\partial x^2}$ con c_s constante y condición inicial $h_2(x, 0) = h(x)$ en terminos de rayos se puede escribir:

al largo del rayo $x_p(t)$ (solución de $\frac{dx}{dt} = c_s, x_p(0) = x_p$)

$\frac{dh_2}{dt} = 0$ (time derivative)

equivalente a $h_2(x_p(t), t) = const = h(x_p)$

De la ecuación de estado del gas perfecto: $T_0 = \frac{p_0}{\rho_0} \frac{\mu}{R}$, donde μ es masa molecular y R la constante de los gases

Ray tracing

Marcamos en el los gráficos un punto $x_p = x_c$ (el centro del paquete) y miramos la evolución en el tiempo desplazandolo en cada paso de tiempo con $c_s dt$. el punto se queda en el centro del paquete. Eso esperabamos porque los rayos son soluciones $x_p(t)$ de $\frac{dx}{dt} = c_g$, $x_p(0) = x_p$ y las amplitudes de las perturbaciones de presión, densidad y velocidad varian lentamente al largo de ellos

en 1D $c_s = c_g$

la ecuacion de la variación de las amplitudes al largo de los rayos es válida para cualquier x_p

Sabemos de la teoria que al largo del rayo $x_p(t)$:

$$k(x_p(t), t) c_s(x_p(t)) = \text{const}$$

Comprobamos la solución numérica:

miramos la evolución en el tiempo de $k_c * c_s(x_p(t))$ que debería ser constante $= k_0 c_s(x_p)$

k_c es el k para cual el valor de los coeficientes fourier $c(k)$ calculados con python de forma numerica con `scipy.fft` para el valor de la presión en el momento t es máximo

pero tiene algunas variaciones como se ve en el gráfico:

k_c no tiene orque ser igual al $k(x_p(t))$ ero si el paquete es bastante estrecho se considera que todos los k estan alrededor de k_c y no pueden variar mucho. Para que esta condición se cumpla $\lambda \ll \text{varLength}$ donde $\text{varLength} = \min_x(|\frac{\rho_0}{\nabla \rho_0}|)(x)$ y λ (wavelength of modulated wave) = `widthPacket`

la condicion $\text{widthPacket} < \text{varLength}$ se cumple solo al principio cuando el paquete era más estrecho

En este caso como se ve en gráfico de arriba $\text{width}(\text{packet})$ crece y $\text{width}(\text{fourier})$ decrece.

La forma gauss del paquete inicial varia muy lentamente (el paquete sigue teniendo una forma (envelope) casi una función gauss se alarga un poco a la derecha si la densidad ρ_0 decrece y se estrecha cuando ρ_0 crece) pero con W más grande y $f_1(k)$ una gauss con w mas pequeño cuando ρ_0 decrece y al revés cuando crece

De la expresión de la transformación fourier esperabamos tener una relación de proporcionalidad inversa entre $\text{width}(\text{packet})$ y $\text{width}(\text{fourier})$

para no usar funciones que calculan de forma empirica la anchura del paquete y de $f_1(k)$

determino la anchura del paquete siguiendo la evolución de 2 puntos (para la presión) que al principio están el borde izquierdo y derecho (los meto con keys "left", "right" en addMarkPoints en model_soundwave.py, pero estos puntos los determino de forma empírica : los defino como $z_c - W * 1.6$, $z_c + W * 1.6$, W siendo el parámetro de la función gauss - definido en sound_wave_packet_params.py)

se que la anchura y el valor maximo de una función gauss son proporcionales

mirando la evolución de $\max(f_1(k))/\text{widthFunction}$ donde widthFunction es la anchura del paquete no parece tan constante (hay que tener en cuenta la evolución lenta de la función gauss que describe la forma del paquete? el periodo de tiempo representado es cuando el paquete pasa por la zona donde la variación de la densidad es mayor y la función se aproxima menos a la gauss):

Amplitudes

De las relaciones de la evolucion teoretica de las amplitudes: en el caso cuando p_0 constante \implies

$|P(x_p(t), t)| c_s^{\frac{1}{2}}(x_p(t), t) = \text{constant}$ al largo del rayo $x_p(t)$

y las otras 2 se calculan de las relaciones : $\frac{|P|}{|V|} = c_s \rho_0$, $\frac{|P|}{|V|} = c_s^2$

Las amplitudes se calculan al largo del rayo $x_p(t)$ (la curva solución para $\frac{dx}{dt} = c_s$, $x(0) = x_p$): C_P es una constante que depende de los valores en el punto inicial del rayo (x_p): $C_P = c_s^{-\frac{1}{2}}(x_p) \rho_0^{\frac{1}{2}}(x_p)$

En la práctica he calculado la solución analítica de $|P|$ al largo del rayo $x_p(t)$:

$$|P(x_p(t), t)| = c_s^{-\frac{1}{2}}(x_p(t)) c_s^{\frac{1}{2}}(x_p) P(x_p, 0)$$

Para ver de forma empirica si paquete se adapta a las curvas de forma $\text{const} * c_s^\alpha$ representamos las curvas:

para la presión: $c_s^{-\frac{1}{2}}(x) c_s^{\frac{1}{2}}(x_0) P(x_0, 0)$

densidad: $c_s^{-\frac{3}{2}}(x) c_s^{\frac{3}{2}}(x_0) R(x_0, 0)$

velocidad: $c_s^{\frac{1}{2}}(x) c_s^{-\frac{1}{2}}(x_0) V(x_0, 0)$

eligiendo x_0 los 2 puntos donde los valores de las amplitudes de las perturbación en el momento inicial tienen el mínimo y máximo

ya sabia que $\alpha = -\frac{1}{2}$ para la presión, $\alpha = -\frac{3}{2}$ (para la densidad, $-\frac{1}{2}$ para rhoCurve), $\alpha = \frac{1}{2}$ para la velocidad (pero se puede variar tambien en el grafico: ver model_soundwave.py)

Conclusiones:

Por la forma que tienen las ecuaciones diferenciales de las variables de cuales queremos estudiar la evolución:

$$\frac{\partial p}{\partial t} + c_s \frac{\partial f}{\partial x} = g(x, t)$$

la solución se puede expresar usando el termino de rayo (creo que se llama de otra forma?)

definimos el rayo $x_p(t)$ la solución de la ecuación $\frac{dx}{dt} = c_s$, $x_p(0) = x_p$

la PDE de arriba se transforma en DE al largo del rayo $x_p(t)$:

$$\frac{dp}{dt} = g(x_p(t), t)$$

k, ω, ϕ, E, a tienen relaciones que se pueden expresar como diferenciales de tiempo al largo de los rayos.

la dependencia de x se transforma en dependencia de tiempo y de x_p el punto inicial del rayo (al momento $t=0$)

$\frac{d\phi}{dt} = 0 \implies$ el paquete no cambia de forma pero su anchura crece(k decrece) cuando c_s crece y se estrecha cuando c_s decrece de forma proporcional a k (por la relación: $\frac{dk}{dt} = -k \frac{dc_s}{dx}$)

de la ecuación diferencial de la amplitud al largo del rayo (o de la energía) se obtiene una relación genérica de dependencia de la amplitud de c_s , ρ_0 , p_0 que tiene que cumplir las 3 variables : perturbación de densidad, presión, velocidad

pero de la ecuación de movimiento de los fluidos y de la condición de adiabaticidad se obtienen otras 2 relaciones entre las amplitudes de cada una

2.3 Documentación usada

http://www.awi.de/fileadmin/user_upload/Research/Research_Divisions/Climate_Sciences/Projekt/1_to_3_WO_waves.pdf

<https://classes.soe.ucsc.edu/ams227/Fall13/lecturenotes/Chapter2-part2.pdf>

2.4 Código

Todo el código está en el repositorio git:

<https://github.com/beevageeva/simnum/>
en la carpeta D1-python

Si no quiere usar git también hay un archivo comprimido de los ficheros *.py en: <https://github.com/beevageeva/simnum/blob/master/python.tar.gz>

Para ejecutar (timeEnd por defecto = timeEnd definido en constants.py):

```
python main.py --timeEnd=5
```

Los parametros de configuración están en:

soundwave_boundary_conditions.py

periodicType = "repeat" — "refl"

soundwave_medium_params.py los parámetros relacionados con el medio : mediumType("homog" o "inhomog")
.p00, v00, rho00 y en el caso del medio inhomogeneo: la funcion de densidad inicial

soundwave_perturbation_params.py los parámetros relacionados a las condiciones iniciales (la perturbación): el tipo de la función y para cada tipo definido aqui hay un fichero distinto donde se configuran los parámetros de cada uno: en el caso del wave packet este fichero es sound_wave_packet_params.py , la amplitud. También se pueden definir superposiciones de funciones.

constants.py:

gamma = 5.0/3, z0, zf, nint, timeEnd, schemeType = "lf" (lax-fr) — "fg" (first generation), fchl, loopType="python" — "weave"

loopType = "weave" \Rightarrow los calculos de las derivadas en alg.py se hacen en C. Tambien para calcular la solución analítica en analytic_solution.py

en el caso de schemeType="fg" se especifica el paso cuando se aplican las condiciones de contorno: en el paso final bcStep = "final" o en el paso intermedio bcStep = "interm"

en notifier_params.py están los parametros relacionados con los gráficos: nStepsPlot = 10 (solo repinta los gráficos cada 10 iteraciones) y si pintar algunos gráficos o no como por ejemplo rhoCurve o fft fourier o fft fourier sol analítica o la solución analítica

la solución analítica no está implementada para cond contorno reflexión

hay mas parámetros que se pueden cambiar que estan la principio de los ficheros python: en visual_plot.py donde están las funciones relacionadas con los gráficos se definen los xlim, ylim, saveImages = True (guarda una imagen en la carpeta outImages_*) y en model_soundwave.py como addMarkPoints que es un hash de los puntos de cuales queremos ver la evolución (en cada iteración los desplazamos con $c_s * dt$) c_s calculado numérico de los valores de la presión y densidad (funciona con superposiciones) .

Para calcular la anchura del paquete hay que añadir a addMarkPoints los puntos que inicial están a los bordes del paquete con los keys: "left" y "right" porque la anchura la calculo siguiendo estos 2 puntos (mejor que con funciones que determinan la anchura de forma empirica: determinando los z cuando la funcion empieza a ser diferente de la derecha y izquierda)