

Estudi numèric dels Shock filters

Adrià Alcalá Mena, David Sánchez Charles

Assignatura de Càlcul Numèric de 4t de la Llicenciatura de Matemàtiques

Abstract—Un dels objectius de l'informàtica és proporcionar-nos eines per a simplificar la nostra vida quotidiana. Un dels problemes que ha de resoldre és el tractament d'imatges: Per exemple, un fons que, amb el pas del temps, s'han deteriorat i volem recuperar part de l'informació, o que les nostres càmeres digitals arreglin els desperfectes que es produeixen a la fotografia per la nostra inexperiència en el maneig d'aquesta eina. Aquests són alguns dels usos que l'informàtica resol i que a les seves solucions es troben unes eines matemàtiques consolidades.

Index Terms—Shock filters, equacions en derivades parcials, càlcul numèric, aproximació numèrica, tractament d'imatges, enfocament.

I. INTRODUCCIÓ

LA passada dècada ha hagut un gran augment de l'utilització d'equacions en derivades parcials dins el camp de la visió per ordinador i el processament d'imatges. S'han desenvolupat aplicacions, amb una rigorosa teoria al darrere, que tenen objectius com la millora d'imatges i eliminació del soroll, segmentació, seguiment d'objectes i molt més. Es pot veure més sobre aquestes aplicacions a [2]. A [1] Osher i Rudin proposen una equació hiperbòlica, anomenada *shock filter* que pot servir com un algorisme estable per a l'enfocament d'imatges aproximant deconvolucions, una convolució és una generalització de fer una mitja ponderada. En el cas d'imatges sol referir-se a que l'imatge s'ha difuminat. Una deconvolució és el procés invers que, matemàticament, és un problema mal posat. Vegem en el cas unidimensional un exemple. En la figura 1 veim la funció signe i el resultat de la convolució amb un nucli de Gauss. L'objectiu dels Shock Filters és partint de la convolució recuperar la funció inicial.

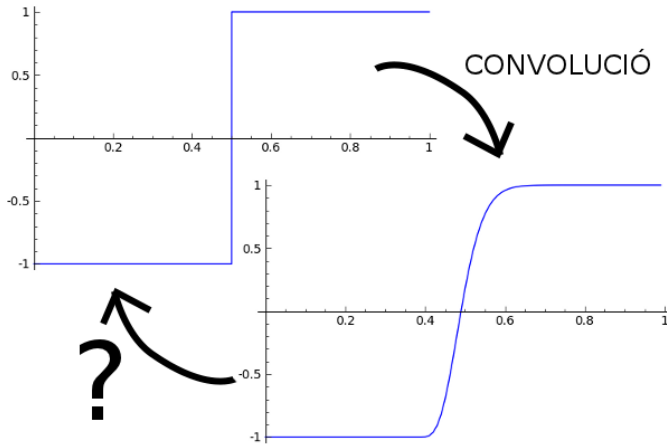


Fig. 1. Aquesta gràfica representa una convolució, el que volem trobar és el mètode que ens retorna la funció original.

II. PLANTEJAMENT DEL PROBLEMA

La formulació de l'equació diferencial del shock filter, proposada per Osher i Rudin, és:

$$u_t = -|u_x|F(u_{xx}), \quad x \in (a, b), t > 0, \quad (1)$$

on F satisfà $F(0) = 0$, $F(s) \cdot \text{sign}(s) \geq 0$. A més, la solució del problema que cercam tindrà com a condicions inicials $u(x, 0) = u_0(x)$ i condicions de frontera de Neumann¹. Si definim $F(s) = \text{sign}(s)$ ens queda l'equació clàssica del shock filter

$$u_t = -|u_x|\text{sign}(u_{xx}). \quad (2)$$

En el cas bidimensional l'equació del shock filter es sol generalitzar amb

$$u_t = -\text{sign}(\Delta u)|\nabla u|, x \in \Omega, t > 0, \quad (3)$$

on ∇u és el gradient² de u i Δu el Laplaciana³ de u .

Algunes propietats generals del shock filter són

- En els punts d'inflexió⁴ el valor de la funció no canvia.
- Els extrems locals no varien al llarg del temps ni se'n creen de nous.
- La solució estacionària és constant a trossos, amb discontinuïtats als punts d'inflexió de u_0 .
- El procés aproxima la deconvolució.

III. CAS UNIDIMENSIONAL

En el cas unidimensional el nostre objectiu és trobar una funció $u : (a, b) \times (0, +\infty) \rightarrow \mathbb{R}$ solució de l'equació diferencial (2) amb condicions inicials $u(x, 0) = u_0(x)$ i satisfent les condicions de frontera Neumann.

El valor de $u(x, T)$ ens proporcionarà l'evolució en l'instant T de la condició inicial $u_0(x)$.

A. Esquema numèric

El mètode numèric que hem fet servir per resoldre aquesta equació diferencial és un mètode que ens permet obtenir una aproximació del valor de la funció en uns punts triats.

Volem obtenir la solució en un interval $[a, b]$, definim $h = \frac{b-a}{M}$ on M serà el nombre de passes desitjades i $x_m = a + m \cdot h$ ($m = 0, \dots, M$) serà la m -èsima passa. Per altra banda, per a la variable temporal escollim uns valors T i N , on T serà

¹ $\frac{\partial u}{\partial n} = 0$ on n és la direcció perpendicular a la frontera

² El vector (u_x, u_y)

³ $u_{xx} + u_{yy}$

⁴ La segona derivada és zero

el temps final que volem calcular i N el nombre d'iteracions que volem fer, i definim $k = \frac{T}{N}$, $t_n = n \cdot k$ ($n = 0, \dots, N$)⁵.

Denotarem per $(u)_m^n$ l'aproximació de la funció u en el punt (x_m, t_n) .

Amb la notació anterior, l'esquema numèric pel cas unidimensional és:

$$u_m^{n+1} = u_m^n - k \cdot |(u_x)_m^n| F((u_{xx})_m^n),$$

on aproximam la primera derivada per

$$(u_x)_m^n = \minmod\left(\frac{u_m^n - u_{m-1}^n}{h}, \frac{u_{m+1}^n - u_m^n}{h}\right). \quad (4)$$

La funció minmod es defineix com

$$\minmod(x, y) = \frac{\text{sign}(x) + \text{sign}(y)}{2} \min\{|x|, |y|\}.$$

Amb la funció *minmod* obtenim una bona aproximació de la derivada u_x en el punt que desitjam:

- Si la derivada té signes diferents en les dues direccions, a prop tenim un mínim/màxim i per tant aproximam la derivada per 0.
- En altre cas, es queda amb la derivada més petita.

Aproximam la segona derivada de u amb l'esquema clàssic de tres punts.

$$(u_{xx})_m^n = \frac{u_{m+1}^n + u_{m-1}^n - 2u_m^n}{h^2}. \quad (5)$$

En ambdues aproximacions podríem tenir problemes en el cas $m = 0$ (i $m = M$) ja que en les fórmules hauríem de conèixer u_{-1}^n (o u_{M+1}^n) que no hem definit. Emprant la condició de frontera

$$\frac{\partial u}{\partial n}(x) = 0 \quad x \in \partial[a, b]$$

que es tradueix per

$$u_x(x) = 0 \quad x = a \text{ o } b$$

i aproximant aquesta derivada amb un esquema de diferències centrades⁶ obtenim que

$$u_{-1}^n = u_1^n \quad u_{M+1}^n = u_{M-1}^n$$

B. Algorisme

L'algorisme per a la resolució numèrica amb aquest esquema es pot resumir en

- 1) $u_m^0 = u_0(x_m)$
- 2) $n = 0$
- 3) $u_m^{n+1} = u_m^n + |(u_x)_m^n| F((u_{xx})_m^n)$ fent ús de les aproximacions esmentades.
- 4) Incrementar n , si hem arribat a $n = N$ param. Si no, tornar a la passa 2.

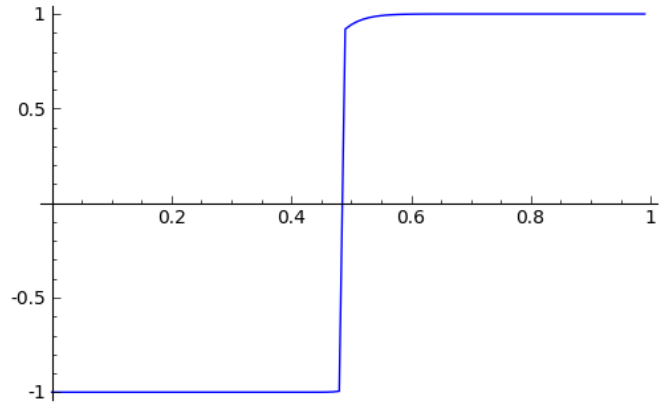


Fig. 2. Shock filter de la convolució de $u(x) = \text{sign}(x)$ ($h = 0.01$ i $k = 0.005$ amb 20 iteracions)

C. Exemples

Vegem primer de tot que passa si a la funció que era la convolució de $u(x) = \text{sign}(x - 0.5)$ (Fig 1) li aplicam el shock filter (Fig 2)

Per tant en un principi, segons podem observar a Fig 2, el shock filter és una deconvolució.

Vegem ara un altre exemple d'aplicar l'algorisme a la funció $u_0(x) = x \cdot \cos x$.

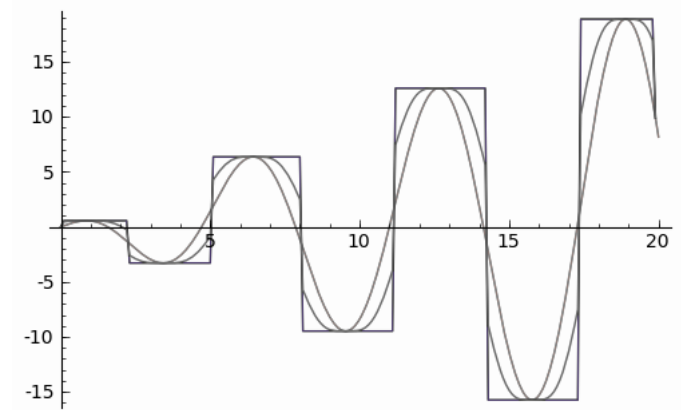


Fig. 3. Shock filter amb $u_0(x) = x \cos(x)$, amb 100 i 2000 iteracions ($h = 0.1$, $k = 0.01$)

Observem a Fig. 3 que als extrems relatius i als punts d'inflexió el valor de la funció no varia. També es pot apreciar que el mètode esglaona la funció conservant aquests punts *conflictius* (extrems relatius i punts d'inflexió).

Fins ara no hem imposat cap condició als valors de h i k , però és necessari que compleixi una certa condició coneguda com *condició CFL*. Podem observar a Fig. 4 que si agafam uns valors no apropiats, el mètode pot produir errors no desitjats. Aquesta condició CFL determina l'estabilitat del mètode, que com es pot veure a [1], el mètode serà estable si $\frac{k}{h} \leq 1/2$

⁵En aquest cas hem fet la partició de l'interval $[0, T]$

⁶ $u_m^n = \frac{u_{m+1}^n - u_{m-1}^n}{h}$

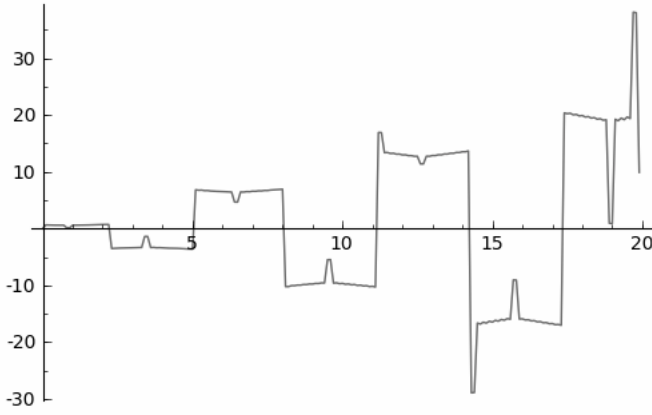


Fig. 4. Shock filter amb $u_0(x) = x \cos(x)$, amb 200 iteracions ($h = 0.1$, $k = 0.2$)

IV. CAS BIDIMENSIONAL

A continuació farem un petit estudi de l'equació del Shock Filter en el cas bidimensional. Sigui $\Omega \subseteq \mathbb{R}^2$ obert, volem trobar una funció $u : \Omega \times (0, +\infty) \rightarrow \mathbb{R}$ solució de l'equació diferencial

$$u_t = -|\nabla u|F(\Delta u), \quad x \in \Omega, t > 0,$$

amb condicions inicials $u(x, y, 0) = u_0(x, y)$ i les condicions de frontera Neumann ⁷.

Aquesta equació diferencial manté les propietats mencionades pel cas general ja que les relacions necessàries entre $|u_x|$ i u_{xx} es mantenen per $|\nabla u|$ i Δu ^{8 9 10}.

El valor de $u(x, y, T)$ ens proporcionarà l'evolució en l'instant T de la funció $u_0(x, y)$.

A. Esquema

Tal com hem fet en el cas unidimensional, farem una partició del nostre domini Ω , que considerarem rectangle. En el nostre cas volem aplicar el mètode sobre una imatge, per tant la discretització de la funció ja ve donada pels píxels de l'imatge. Només ens queda escollir el temps final, T , i el nombre d'iteracions, N , que volem calcular.

En aquest cas denotam per $u_{i,j}^n$ el nivell de gris del píxel (i, j) en el temps $t_n = n \cdot \frac{T}{N}$ ($n = 0, \dots, N$).

Amb aquesta notació el nostre esquema és:

$$u_{i,j}^{n+1} = u_{i,j}^n - |(\nabla u)_m^n| F((\Delta u)_m^n).$$

Recordam que

$$|\nabla u| = \sqrt{u_x^2 + u_y^2} \quad \Delta u = u_{xx} + u_{yy},$$

on, per aproximar aquests valors, farem ús de les mateixes fórmules (4) i (5) utilitzades en el cas unidimensional, per aproximar u_x , u_y , u_{xx} i u_{yy} .

⁷ $\frac{\partial u}{\partial n}(x, y) = 0, (x, y) \in \partial\Omega$

⁸ En tot mínim/màxim, $|\nabla u| = 0$

⁹ En tot mínim, $\Delta u > 0$

¹⁰ En tot màxim, $\Delta u < 0$

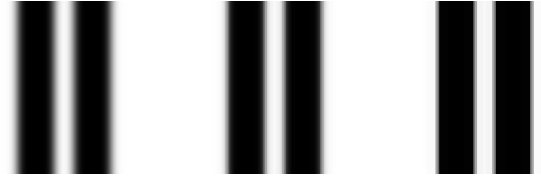


Fig. 5. D'esquerra a dreta: Imatge original, $T = 0.2$ i $N = 4$, $T = 1.8$ i $N = 36$

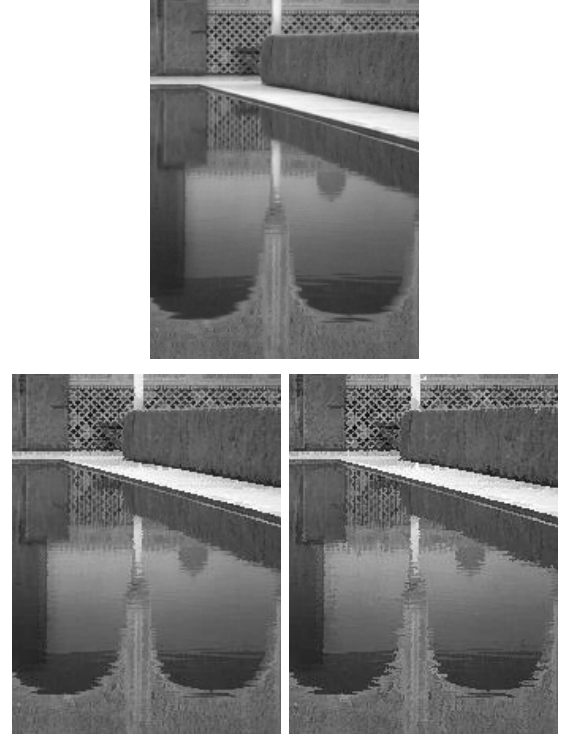


Fig. 6. A dalt tenim la imatge original. A l'esquerra la imatge processada amb 4 iteracions fins a arribar al temps $T = 0.2$. A la dreta després de 36 iteracions per a arribar al temps $T = 1.8$.

B. Exemples

Vegem com es comporta aquest mètode sobre imatges. Primer ho farem sobre una imatge artificial per a veure el seu comportament bàsic. Totes les proves s'han fet amb un increment $h = 0.1$ per a la variable espacial (tant en la component X com en la Y). Recordem que és necessari que els valors de h i k satisfacin la condició CFL que, en el cas bidimensional, és $\frac{k}{h} \leq 1/4$, ja que $F(x) = \text{sign}(x)$ [1]

A Fig. 5 podem observar el mateix efecte que es produeix a la Fig. 3 però sobre una imatge. El mètode esglaona aquesta imatge borrosa fins que recuperam, bantant bé, uns rectangles ben definits.

Anem a veure com funciona el mètode sobre imatges reals. Primer hem fet ús d'aquest mètode per a veure com es comporta sobre el reflexe en aigua (Fig. 6)

Vegem ara que passa si aplicam el mètode a una imatge clàssica dins el tractament d'imatges després d'haver-la desenfocat (Fig 7). Notem que hem aconseguir part del nostre objectiu, les vores de la imatge es veuen més nítides. Però, hem pagat un preu per aquesta nitidesa, i és que les vores comencen

a pixelar-se molt. En l'exemple on la imatge *original* és molt més borrosa, encara que el mètode aconsegueix realçar les seves vores, obtenim un resultat més artificial, quasi pareix un dibuix.



Fig. 7. A l'esquerra tenim dos emborronaments de l'imatge original, i a la dreta tenim la imatge després de 40 iteracions amb $k = 0.025$.

V. CONCLUSIONS

La poca quantitat d'equacions diferencials que tenen solució analítica ha fet que els matemàtics comencessin a desenvolupar eines per a obtenir una aproximació numèrica d'aquestes ja en el segle XVIII: Leonhard Euler (1707-1783) inventà el primer mètode per resoldre'n, època en que era impensable l'existència d'unes màquines que ens poguessin executar aquests mètodes amb rapidesa. Ara que ja tenim una maquinària suficientment potent, els matemàtics, i no matemàtics, s'han donat compte de la potència de la resolució numèrica d'equacions diferencials. No només pel tractat en aquest article, sinó en innombrables escenaris com poden ser el càlcul de la trajectòria de satèl·lits, modelatge del creixement d'una població o de la oferta i demanda d'un producte, etc.

AGRAÏMENTS

Amb la col·laboració de la Dra. Catalina Sbert Juan, doctora en Informàtica per la Universitat de les Illes Balears

REFERENCES

- [1] L. I. Rudin, S. Osher, *Feature-Oriented Image Enhancement with Shock Filters*, SIAM J. Numer. Anal. 27, pp. 919-940, 1990
- [2] G. Gilboa, N.A. Sochen, and Y.Y. Zeevi, *Regularized Shock Filters and Complex Diffusion*, in Proc. ECCV (1), 2002, pp.399-413.