

Solución del Problema de Valores Iniciales de la ecuación de Black-Scholes usando Funciones de Base Radial y algoritmos de Sumación Rápida.

Juan Antonio Licea, CIMAT
Marcos Aurelio Capistrán, CIMAT

ENOAN 2010

11 de marzo de 2010

Introducción:

Objeto de estudio:

El Problema de Valores Iniciales de la Ecuación de Black-Scholes via funciones de base radial.

Problema de Valores Iniciales de Black-Scholes (PVIBS).

$$\frac{\partial V}{\partial t} = r(t)V - r(t)S \frac{\partial V}{\partial S} - \frac{1}{2} \sigma^2(S, t) S^2 \frac{\partial^2 V}{\partial S^2},$$

$$V(S, T) = V_T(S)$$

$$S \in \mathbb{R}_+, t \in [0, T]$$

Objetivos:

Aproximar la solución de este problema mediante interpolación con Funciones de Base Radial (FBR).

Objetivos:

Aproximar la solución de este problema mediante interpolación con Funciones de Base Radial (FBR).

Para esto utilizamos herramientas estándar:

- Algoritmos iterativos para interpolación con FBR.
- Algoritmos de sumación rápida de FBR.

Plan de la Charla:

- 1 Introducción**
 - Problema de Valores Iniciales de Black-Scholes
- 2 Marco Teórico**
 - Estado del Arte
- 3 Metodología**
 - Discretización
 - Algoritmos iterativos
 - Algoritmos de sumación rápida de FBR
- 4 Resultados**
- 5 Conclusiones**

El PVIBS es un problema clásico:

Una **EDP Parabólica** con **datos finales de Cauchy**.

Existencia y unicidad de soluciones

Teorema:

Sea $\Omega = \mathbb{R}^+ \times [0, T]$, si:

- $S\sigma(S, t)$ es Lipschitz en Ω .
- $\sigma(S, t)$ es acotada en Ω y acotada por abajo por una constante positiva.
- $r(t)$ es acotada y Lipschitz.
- Los datos de Cauchy $V_T(S)$ satisfacen $0 \leq V_T(S) \leq C(1 + S)$ para C una constante dada.

Entonces existe una única solución $V \in \mathcal{C}^0(\Omega)$, \mathcal{C}^1 con respecto de t y \mathcal{C}^2 con respecto de S , tal que $0 \leq V(S, t) \leq C'(1 + S)$ para C' una constante dada.

Métodos Estándar de Solución

Se han estudiado varios métodos numéricos para resolver el PVIBS

- diferencias finitas
- elementos finitos
- volúmenes finitos

Métodos Estándar de Solución

Se han estudiado varios métodos numéricos para resolver el PVIBS

- diferencias finitas
- elementos finitos
- volúmenes finitos

Una alternativa:

Interpolación con Funciones de Base Radial.

Por qué Funciones de Base Radial?

- Métodos libres de malla
- Implementación similar en cualquier dimensión

Por qué Funciones de Base Radial?

- Métodos libres de malla
- Implementación similar en cualquier dimensión

Además, desde el punto de vista de las aplicaciones, es útil calcular las derivadas parciales de la solución de este problema.

Discretización espacial

Aproximamos $V(S, t)$ por Funciones de Base Radial:

$$V(S, t) \approx v(S, t) = \sum_{j=1}^N \alpha_j(t) \Phi(S - S_j) + \lambda(t)$$

- $\Phi(S) = \phi(|S|)$ son las FBR (ϕ una función dada).
- $\alpha_j(t)$ son los coeficientes.

Sustituyendo $v(S, t)$ en el PVIBS, se obtiene la EDO:

$$\begin{aligned}\mathbf{L}\dot{\alpha} &= \mathbf{P}\alpha \\ \mathbf{L}\alpha(T) &= \mathbf{V}_T\end{aligned}$$

- $\mathbf{P} = r\mathbf{L} - \mathbf{L}_S - \frac{1}{2}\sigma^2\mathbf{L}_{SS},$
- $\alpha = [\alpha_j(t)],$
- $\mathbf{L} = [\Phi(S_i - S_j)],$
- $\mathbf{L}_S = [S_i\Phi_S(S_i - S_j)],$
- $\mathbf{L}_{SS} = [S_i^2\Phi_{SS}(S_i - S_j)].$

Resolver un sistema lineal de Ecuaciones Diferenciales Ordinarias.

- Euler
- Runge Kutta
- Métodos implícitos CN, **BDF2** [3]

Discretización temporal

Método implícito BDF2:

$$\alpha(T) = \mathbf{L}^{-1} \mathbf{V}_T$$

$$\alpha^1 = (\mathbf{L} + \Delta t_1 \mathbf{P})^{-1} \mathbf{L} \alpha^0$$

$$\alpha^n = (\mathbf{L} + \beta_0^n \mathbf{P})^{-1} \mathbf{L} (\beta_1^n \alpha^{n-1} - \beta_2^n \alpha^{n-2}), \quad n = 2, \dots, M$$

► coeficientes

Costo computacional:

| Operaciones | Cantidad & costo |
|---|------------------|
| | $O(N^3)$ |
| \mathbf{L}^{-1} | 1 |
| $(L + \Delta t_1 \mathbf{P})^{-1}$ | 1 |
| $(L + \beta_0^n \mathbf{P})^{-1}$ | 1 |
| | $O(N^2)$ |
| $\mathbf{L}^{-1} \mathbf{V}_T$ | 1 |
| $(L + \Delta t_1 \mathbf{P})^{-1} \mathbf{L} \alpha^0$ | 1 |
| $(\mathbf{L} + \beta_0^n \mathbf{P})^{-1} \mathbf{L} (\beta_1^n \alpha^{n-1} - \beta_2^n \alpha^{n-2})$ | $M - 2$ |
| Total | $O(N^3 + MN^2)$ |

Invertir matrices es una tarea muy costosa, es preferible resolver los siguientes sistemas de ecuaciones de manera eficiente:

Método implícito BDF2:

$$\mathbf{L}\boldsymbol{\alpha}(T) = \mathbf{V}_T$$

$$(\mathbf{L} + \Delta t_1 \mathbf{P})\boldsymbol{\alpha}^1 = \mathbf{L}\boldsymbol{\alpha}^0$$

$$(\mathbf{L} + \beta_0^n \mathbf{P})\boldsymbol{\alpha}^n = \mathbf{L}(\beta_1^n \boldsymbol{\alpha}^{n-1} - \beta_2^n \boldsymbol{\alpha}^{n-2}), \quad n = 2, \dots, M$$

Resolver sistemas de la forma:

$$\tilde{\mathbf{P}}\alpha^{n+1} = \tilde{\alpha}^n$$

α_n - discretización de los coeficientes.

Si se utiliza un método iterativo para resolver sistemas de ecuaciones lineales

Costo computacional:

| Operaciones | Costo | Cantidad |
|--|------------|---------------------|
| resolver $\tilde{\mathbf{P}}\alpha^{n+1} = \tilde{\alpha}^n$ | $O(kN^2)$ | M |
| Total | $O(kMN^2)$ | vs. $O(N^3 + MN^2)$ |

k - número de iteraciones necesarias resolver $\mathbf{L}\alpha^n = \tilde{\alpha}$.

Nota:

También es posible agregar un método para evaluar de manera eficiente los productos matriz vector requeridos.

Nota:

Los sistemas que surgen en problemas de interpolación por medio de FBR generalmente son mal condicionados.

Precondicionamiento con funciones cardinales aproximadas

La función cardinal asociada al punto x_j es una función de la forma

$$z^l(x) = \sum_{j=1}^N \zeta_j^l \Phi(x - x_j) + \lambda_l$$

que satisface las condiciones de Kronecker

$$z^l(x_j) = \delta_{lj}$$

Precondicionamiento con funciones cardinales aproximadas

La función cardinal asociada al punto x_j es una función de la forma

$$z^l(x) = \sum_{j=1}^N \zeta_j^l \Phi(x - x_j) + \lambda_l$$

que satisface las condiciones de Kronecker

$$z^l(x_j) = \delta_{lj}$$

Nota:

Si se conocen los coeficientes de estas funciones, es fácil recobrar la solución.

- Aproximar la función cardinal z^l correspondiente a x_l utilizando un conjunto pequeño de puntos \mathcal{L}_l .

$$\hat{z}^l(x) = \sum_{j \in \mathcal{L}_l} \zeta_j^l \Phi(x - x_j) + \lambda_l$$

- Aproximar la función cardinal z^l correspondiente a x_l utilizando un conjunto pequeño de puntos \mathcal{L}_l .

$$\hat{z}^l(x) = \sum_{j \in \mathcal{L}_l} \zeta_j^l \Phi(x - x_j) + \lambda_l$$

- Utilizar la matriz

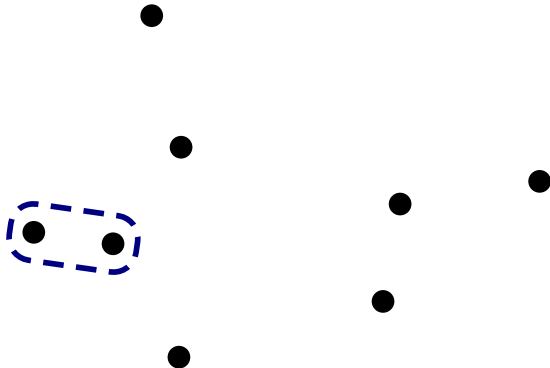
$$\hat{T} = [\zeta_l^j]$$

como preconditionador en un algoritmo del tipo Gradiente Conjugado. [▶ Formulas \[7\]](#)

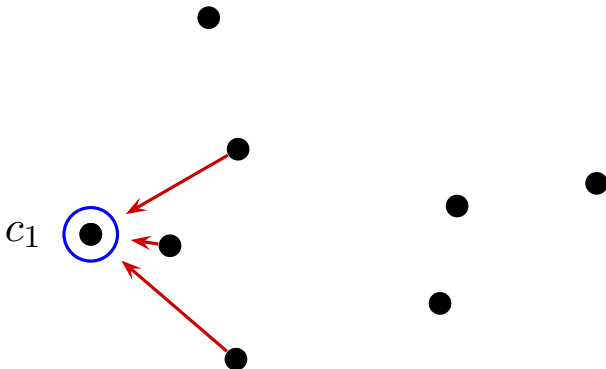
Costo de construir el preconditionador \hat{T}

- Calcular conjuntos de puntos especiales para aproximar \hat{z}^l :
 $O(N^2)$ [8].
- Calcular los coeficientes ζ_j^l de \hat{T} :
 $O(Nq^3)$, para $q \ll N$.

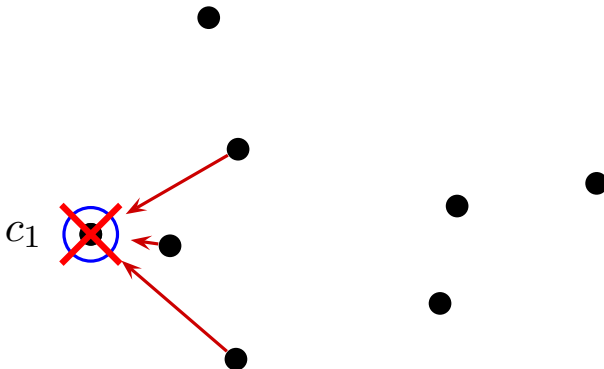
Puntos: Ejemplo



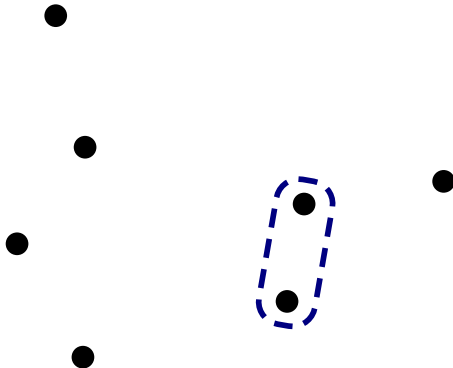
Puntos: Ejemplo



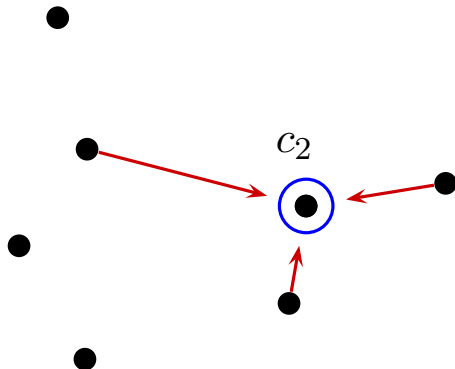
Puntos: Ejemplo



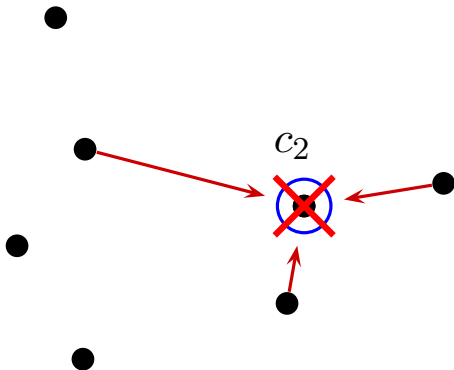
Puntos: Ejemplo



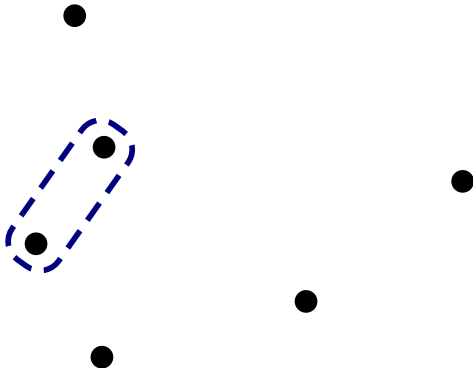
Puntos: Ejemplo



Puntos: Ejemplo



Puntos: Ejemplo



Suma rápida de FBR multicuádricas:

Requerimos evaluar de manera eficiente sumas de la forma:

$$s(y) = \sum_{i=1}^N \lambda_i (\|y - x_i\|^2 + c^2)^{k/2},$$

en N puntos distintos.

“La evaluación directa requiere $O(N^2)$ operaciones.”

Evaluar $A\lambda$:

$$A = [y_j x_i], \quad \lambda = [\lambda_i]$$

Evaluar $A\lambda$:

$$\mathbf{A} = [y_j x_i], \quad \boldsymbol{\lambda} = [\lambda_i]$$

$$s(y_j) = \sum_{i=1}^N \lambda_i y_j x_i = y_j \sum_{i=1}^N \lambda_i x_i$$

Evaluar $A\lambda$:

$$\mathbf{A} = [y_j x_i], \quad \boldsymbol{\lambda} = [\lambda_i]$$

$$s(y_j) = \sum_{i=1}^N \lambda_i y_j x_i = y_j \sum_{i=1}^N \lambda_i x_i$$

Evaluar en 2 etapas:

- 1 Precalcular $c = \sum_{i=1}^N \lambda_i x_i$
- 2 Evaluar $s(y_j) = c y_j$

Evaluar $A\lambda$:

$$A = [y_j x_i], \quad \lambda = [\lambda_i]$$

$$s(y_j) = \sum_{i=1}^N \lambda_i y_j x_i = y_j \sum_{i=1}^N \lambda_i x_i$$

Evaluar en 2 etapas:

- 1 Precalcular $c = \sum_{i=1}^N \lambda_i x_i$
- 2 Evaluar $s(y_j) = c y_j$

Que pasa si ϕ no es un polinomio?

Métodos rápidos (Aproximados):

- Método multipolos
- Suma basada en la Transformada Rápida de Gauss.

Método Multipolos

Introducido por Greengard y Rokhlin, 1987 [2].

Descripción:

- 1 División del dominio (estructura de árbol).
- 2 Expansion de $\phi(|x - y|)$ en serie de Laurent.
- 3 Aproximar $s(x)$ mediante un método jerárquico.

Expansiones [1]

Si $t_i \in \mathbb{R}^n$, $|y_i| \leq r$, y $d_i \in \mathbb{R}$ para $1 \leq i \leq N$, k impar, entonces

$$Q_l(x) = \sum_{i=0}^N d_i P_l^k(|y_i|^2 + \tau^2, -2 \langle y_i, x \rangle, |x|^2),$$

$$s_p(x) = \sum_{l=0}^{p+k} Q_l(x) \frac{1}{|x|^{2l-k}}, \text{ donde}$$

$$P_l^{(k)}(u, v, w) = \sum_{j=\lfloor \frac{l+1}{2} \rfloor}^l \binom{k/2}{j} \binom{j}{l-j} v^{2j-l} (uw)^{l-j}, \quad l \geq 0.$$

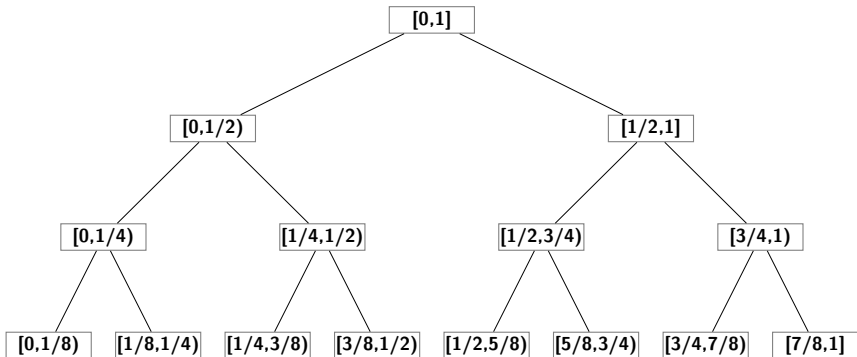
► cota

Traslación

$$s_p(y) \approx \sum_{l=0}^{p+k} \left(\sum_{j=0}^l q_j(y) P_{l-j}^{(-2p-k)}(|t|^2, -2\langle y, t \rangle, |y|^2) \right) \frac{1}{|y|^{2l-k}}$$

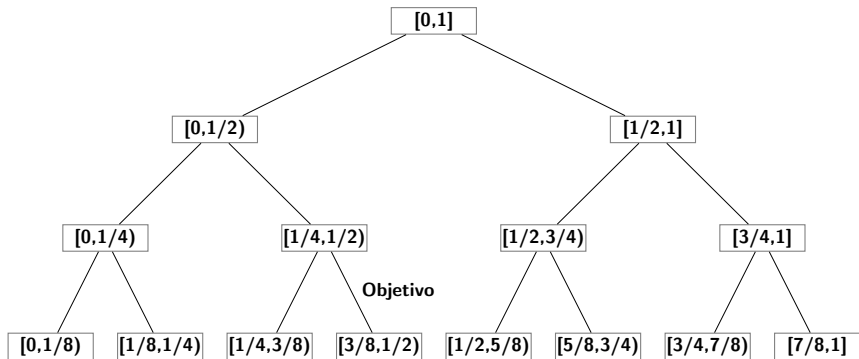
Método Multipolos

Paso 1: Dividir el dominio y calcular expansiones



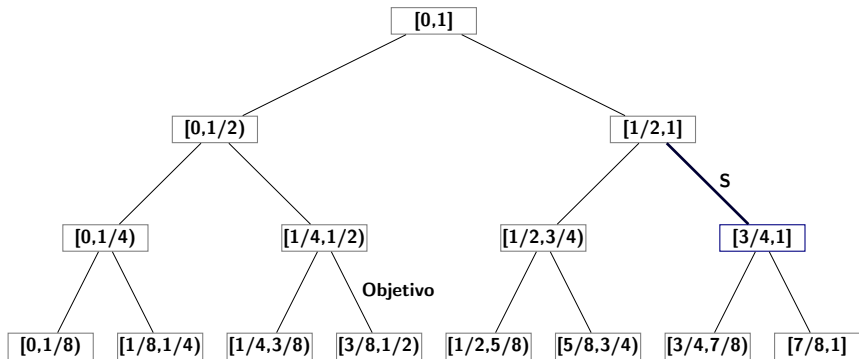
Método Multipolos

Paso 2: Suma aproximada (NIVEL 2)



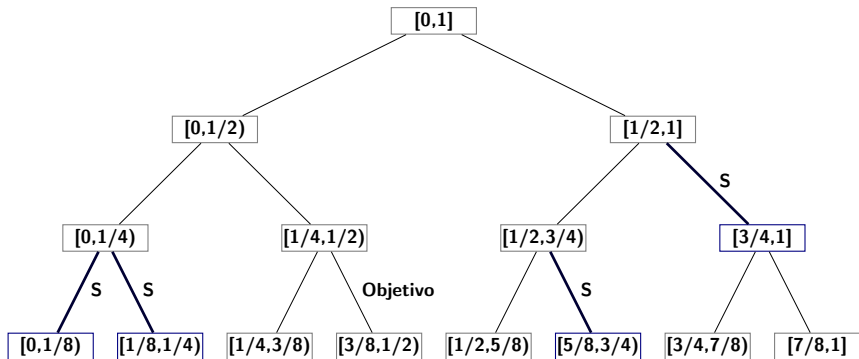
Método Multipolos

Paso 2: Suma aproximada (NIVEL 3)



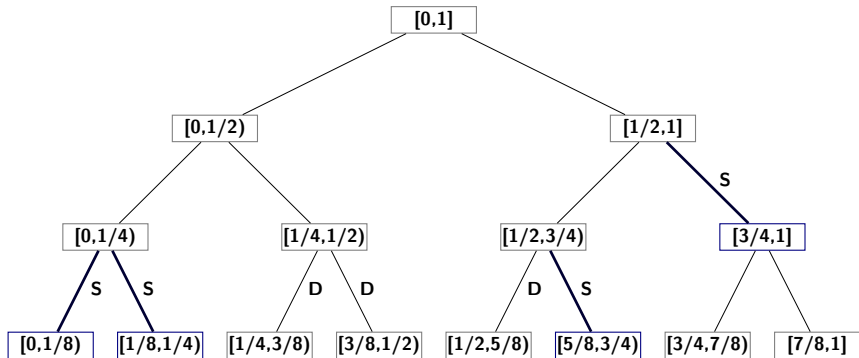
Método Multipolos

Paso 2: Suma aproximada (NIVEL 4)



Método Multipolos

Paso 2: Suma directa (NIVEL 4)



Formulas para traslación de expansiones, caso unidimensional

En este caso se puede desacoplar

$$s_p(y) = \text{sign}(y - t) \sum_{l=0}^{p+k} \frac{Q_l}{(y - t)^{l-k}},$$

$$Q_l = \sum_{i=1}^N \lambda_i P_l^{(k)}((x_i - t)^2 + c^2, -2(x_i - t), 1).$$

Para $k = 1$ la serie centrada en t_1 es

$$r_T(y) = \text{sign}(y - t_1) \sum_{l=0}^{p+1} \frac{Q_l}{(y - t_1)^{l-1}}, \quad dt = t_2 - t_1$$

requerimos escribir la expansión en términos de $y - t_2$.

Usar la fórmula:

$$\frac{1}{(y - t_1)^{l+1}} = \sum_{m=l}^{\infty} (-1)^{m-l} \binom{m}{l} dt^{m-l} \frac{1}{y - t_2}.$$

Usar la fórmula:

$$\frac{1}{(y - t_1)^{l+1}} = \sum_{m=l}^{\infty} (-1)^{m-l} \binom{m}{l} dt^{m-l} \frac{1}{y - t_2}.$$

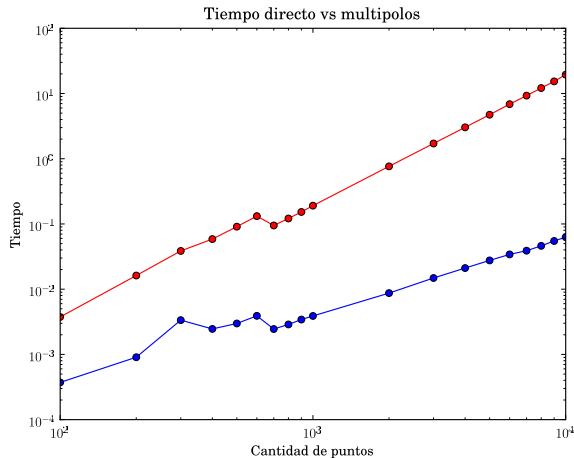
La serie trasladada está dada por:

$$r_T(y) = \text{sign}(y - t_2) \sum_{l=0}^{p+1} \frac{\tilde{Q}_l}{(y - t_2)^{l-1}}, \text{ donde}$$

$$\tilde{Q}_0 = Q_0$$

$$\tilde{Q}_1 = Q_0 dt + Q_1$$

$$\tilde{Q}_l = \sum_{m=0}^l (-1)^{m-l} \binom{m}{l} dt^{m-l} Q_l, \quad l \geq 2.$$



Ejemplo:

Suma de FBR
multicuádricas con
el método
multipolos

Datos:

- Puntos en $[0,1]$.
- 10 términos.
- $c = 1/N$
- $k = 1$

Suma de multicuádricas basada en la TRGM

Introducido por Roussos y Baxter, 2005 [4].

Descripción:

- 1 Representación integral de $\phi(|x - y|)$
(que contiene un término exponencial).
- 2 Integración numérica - Suma de Gaussianas.

Representación integral de ϕ

Multicuádrica

$$\begin{aligned} & \sum_{i=1}^N \lambda_i \sqrt{\|y - x_i\|^2 + c^2} \\ &= c + \frac{1}{2\sqrt{\pi}} \int_0^\infty \frac{e^{-sc^2}}{\sqrt{s}} \left(\frac{\sum_{i=1}^N \lambda_i - \sum_{i=1}^N \lambda_i e^{-s\|y-x_i\|^2}}{s} \right) ds \end{aligned}$$

Multicuádrica Inversa

$$\sum_{i=1}^N \frac{\lambda_i}{\sqrt{\|y - x_i\|^2 + c^2}} = \frac{1}{\sqrt{\pi}} \int_0^\infty \frac{e^{-sc^2}}{\sqrt{t}} \sum_{i=1}^N \lambda_i e^{-s\|y-x_i\|^2} ds$$

Integración numérica

$$s(y) \approx \frac{1}{\sqrt{\pi}} \sum_{k=1}^q w_k f(s_k)$$

$$f(s) = \sum_{i=1}^N \lambda_i e^{-s \|y - x_i\|^2}$$

Integración numérica

$$s(y) \approx \frac{1}{\sqrt{\pi}} \sum_{k=1}^q w_k f(s_k)$$

$$f(s) = \sum_{i=1}^N \lambda_i e^{-s \|y - x_i\|^2}$$

Trabajo:

Sumar q Gaussinas en $O(N)$ operaciones mediante la Transformada Rápida de Gauss [5], en lugar de una Multicuádrícula a un costo $O(N^2)$.

Transformada Rápida de Gauss Mejorada (TRMG) [6]

Descripción:

- Serie de Taylor de la FBR exponencial.
- División del dominio (k centros).
- Sólo se suman contribuciones en una vecindad.

Factorización clave:

$$e^{-\|y_j - x_i\|^2/h^2} = e^{-\|\Delta y_j\|^2/h^2} e^{-\|\Delta x_i\|^2/h^2} e^{2\Delta y_j \cdot \Delta x_i/h^2}$$

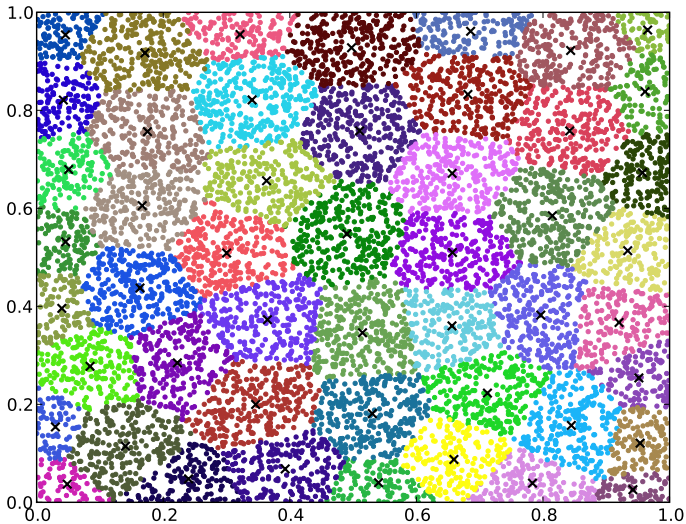
$$e^{2x \cdot y} = \sum_{\alpha \geq 0} \frac{2^{|\alpha|}}{\alpha!} x^\alpha y^\alpha$$

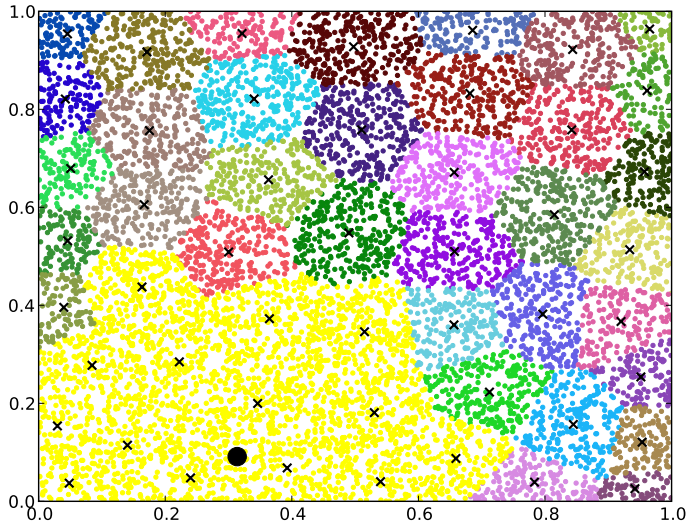
$$\Delta y_j = y_j - y^*, \quad \Delta x_i = x_i - x^*$$

Expansiones:

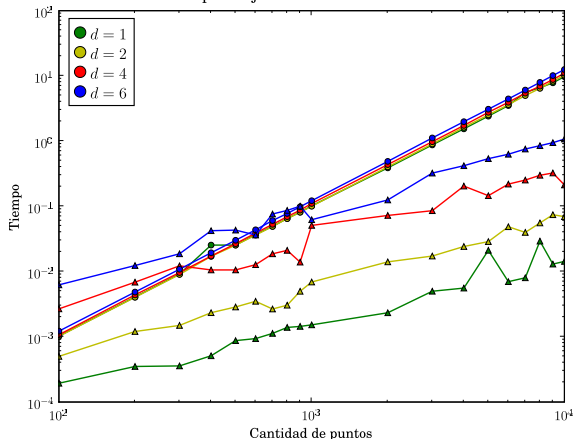
$$G(y_j) = \sum_{\|y_j - c_k\| \leq h\rho_y} \sum_{|\alpha| < p} C_\alpha^k e^{-\|y_j - c_k\|^2/h^2} \left(\frac{y_j - c_k}{h} \right)^\alpha$$

$$C_\alpha^k = \frac{2^{|\alpha|}}{\alpha!} \sum_{x_i \in S_k} q_i e^{-\|x_i - c_k\|^2/h^2} \left(\frac{x_i - c_k}{h} \right)^\alpha$$





Tiempo de ejecución: TRGM vs. Directo.

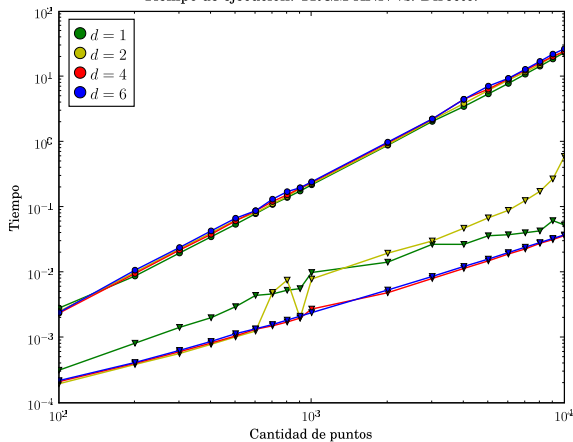
**Ejemplo:**

Suma de FBR
Gaussianas con la
TRGM

Datos:

- Puntos en el hipercubo unitario
- $h = d/2$

Tiempo de ejecución: TRGM-ANN vs. Directo.



Ejemplo:

Suma de FBR
Gaussianas con la
TRGM-ANN [10]

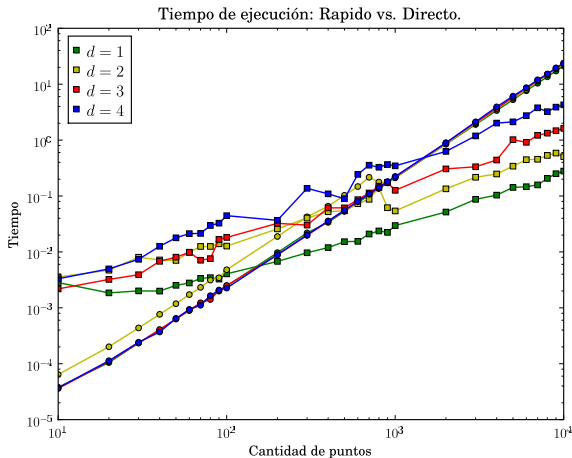
Datos:

- Puntos en el hipercubo unitario
- $h = 0.01d$

Nota:

Siguiendo la idea de Roussos, se puede utilizar la siguiente fórmula para la evaluación rápida de FBR multicuádricas con exponente negativo arbitrario

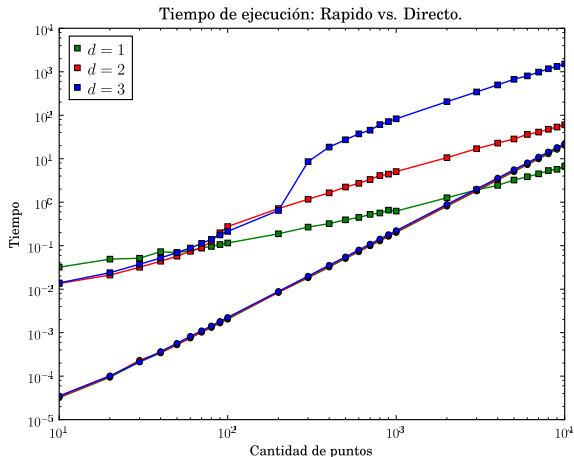
$$(\|x\|^2 + c^2)^{\nu/2} = \frac{1}{\Gamma\left(\frac{-\nu}{2}\right)} \int_0^\infty \frac{e^{-sc^2}}{s^{\nu/2+1}} e^{-s\|x\|^2} ds, \quad \nu < 0$$

**Ejemplo:**

Suma de FBR
Multicuadradas con
la TRGM

Datos:

- Puntos en el hipercubo unitario
- $c = 2d$



Ejemplo:

Suma de FBR
Multicuadradas con
la TRGM

Datos:

- Puntos en el hipercubo unitario
- $c = 0.2d$

Evaluación de $P = rL - rL_s - \frac{1}{2}\sigma^2L_{ss}$ ▶ spline

Evaluación de $\mathbf{P} = r\mathbf{L} - r\mathbf{L}_s - \frac{1}{2}\sigma^2\mathbf{L}_{ss}$ ▶ spline

$$\begin{aligned} [\mathbf{P}\boldsymbol{\lambda}]_i &= r \sum_{j=1}^N \lambda_j \left((s_i - s_j)^2 + c^2 \right)^{1/2} \\ &\quad - r s_i \sum_{j=1}^N \lambda_j (s_i - s_j) \left((s_i - s_j)^2 + c^2 \right)^{-1/2} \\ &\quad - \frac{1}{2} (c\sigma s_i)^2 \sum_{j=1}^N \lambda_j \left((s_i - s_j)^2 + c^2 \right)^{-3/2} \end{aligned}$$

Evaluación de $P = rL - rL_s - \frac{1}{2}\sigma^2L_{ss}$

$$\begin{aligned} [\mathbf{P}\boldsymbol{\lambda}]_i &= r \sum_{j=1}^N \lambda_j \left((s_i - s_j)^2 + c^2 \right)^{1/2} \\ &\quad - r s_i^2 \sum_{j=1}^N \lambda_j \left((s_i - s_j)^2 + c^2 \right)^{-1/2} \\ &\quad + r s_i \sum_{j=1}^N \lambda_j s_j \left((s_i - s_j)^2 + c^2 \right)^{-1/2} \\ &\quad - \frac{1}{2} (c\sigma s_i)^2 \sum_{j=1}^N \lambda_j \left((s_i - s_j)^2 + c^2 \right)^{-3/2} \end{aligned}$$

$M(\boldsymbol{\lambda}, \mathbf{s}, k) \equiv$ suma con exponente k , coeficientes $\boldsymbol{\lambda} = [\lambda_i]$ y puntos $\mathbf{s} = [s_i]$

$$\begin{aligned} [\mathbf{P}\boldsymbol{\lambda}]_i &= rM(\boldsymbol{\lambda}, \mathbf{s}, 1) \\ &\quad -rs^2 * M(\boldsymbol{\lambda}, \mathbf{s}, -1) \\ &\quad +rs * M(\boldsymbol{\lambda} * \mathbf{s}, \mathbf{s}, -1) \\ &\quad -\frac{1}{2}(\sigma c)^2 \mathbf{s}^2 * M(\boldsymbol{\lambda}, \mathbf{s}, -3) \end{aligned}$$

Nos interesa evaluar:

$$\mathbf{P}_0 = \mathbf{L}$$

$$\mathbf{P}_1 = \mathbf{L} + \Delta t_1 \mathbf{P}$$

$$\mathbf{P}_2 = \mathbf{L} + \beta \mathbf{P}$$

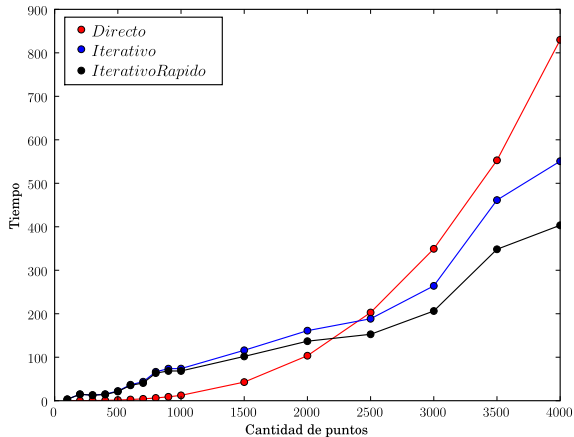
Marcha en el tiempo

Denotemos por $\alpha, \lambda = \text{FGP05}_{(\cdot)}(y)$ la solución de $P_{(\cdot)}\lambda + \alpha$

$$\begin{aligned} \mathbf{y}_0 &= \mathbf{V}_T \\ \alpha^0, \lambda^0 &= \text{FGP05}_0(\mathbf{y}_0) \end{aligned}$$

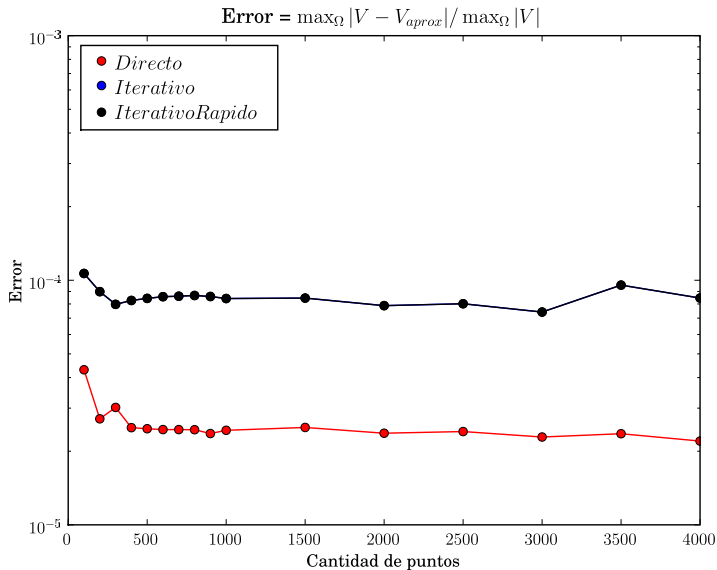
$$\begin{aligned} \mathbf{y}_1 &= \mathbf{L}\alpha^0 + \lambda^0 \\ \alpha^1, \tilde{\lambda}^1 &= \text{FGP05}_1(\mathbf{y}_1) \\ \lambda^1 &= (1 + \Delta t_1)\tilde{\lambda}^1 \end{aligned}$$

$$\begin{aligned} \mathbf{y}_n &= \mathbf{L}(\beta_1^n \alpha^{n-1} - \beta_2^n \alpha^{n-2}) + \beta_1^n \lambda^{n-1} - \beta_2^n \lambda^{n-2} \\ \alpha^n, \tilde{\lambda}^n &= \text{FGP05}_2(\mathbf{y}_n) \\ \lambda^n &= (1 + \beta)\tilde{\lambda}^n, \quad n = 2, \dots, M \end{aligned}$$



Ejemplo:

- $call$
- $K = 10$
- $r = 0.05$
- $\sigma = 0.2$
- $T = 0.5$



Comentarios.

Se desarrolla un esquema para aproximar la solución de un problema de valores iniciales para una EDP parabólica utilizando interpolación con funciones de base radial. Se incorporan dos herramientas estándar

- Un algoritmo iterativo
- Evaluación eficiente de FBR multicuádricas