

**CIMAT**

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

Tesis presentada por:

Juan Antonio Licea Salazar

Maestría en Ciencias con especialidad en Matemáticas Aplicadas.

Centro de Investigación en Matemáticas A.C.

2009

Asesor: Dr. Marcos Aurelio Capistrán Ocampo



*A mi familia.*

# Agradecimientos

Agradezco a mi asesor el Dr. Marcos Aurelio Capistrán por proporcionarme apoyo, consejo y motivación a lo largo del desarrollo de esta tesis y al Dr. Miguel Ángel Moreles por haber impartido un curso que fué de gran ayuda para realizar este trabajo.

También me gustaría agradecer a las instituciones que me brindaron apoyo para completar mis estudios de posgrado:

- El Centro de Investigación en Matemáticas A.C.
- El Consejo Nacional de Ciencia y Tecnología que me otorgó la beca para estudios de posgrado.
- El Consejo de Ciencia y Tecnología del Estado de Guanajuato, por haberme proporcionado una beca para tesis de Maestría.

No podría estar más agradecido con mi esposa e hija, mi madre, mi padre y hermanos por todo el apoyo que me brindaron, y siempre me motivaron para seguir adelante.

# Resumen

El problema de asignar un precio a las opciones financieras de una canasta de bienes se puede formular como un problema de Cauchy para una ecuación diferencial parcial parabólica (Black-Scholes) en varias variables. En este trabajo proponemos aproximar la solución de este problema mediante interpolación con funciones de base radial. Para esto usamos herramientas estándar tales como algoritmos iterativos para interpolación con funciones de base radial y métodos de sumación rápida. Nos restringimos a funciones de base radial multicuádricas. Se implementaron varios algoritmos estándar y se obtuvo evidencia numérica de que la estrategia propuesta es apropiada para resolver el problema de valores iniciales que nos ocupa. Encontramos que aunque los sistemas de ecuaciones lineales resultantes son densos, las matrices correspondientes tienen pocos grados de libertad, es decir, los problemas de interpolación y de multiplicar una matriz por un vector resultantes son intrínsecamente de bajo costo computacional.

El método multipolos es una opción para evaluar los productos matriz vector en el caso unidimensional. Sin embargo, como es conocido, la eficiencia de este método decae conforme aumenta la dimensión, y se requiere un esfuerzo considerable para obtener una implementación eficiente en dimensiones mayores a 1 (por tal razón en este trabajo sólo se implementó el caso unidimensional).

Aparentemente el método rápido para la suma de funciones multicuádricas propuesto en [18] en combinación con la Transformada Rápida de Gauss Mejorada [11] es una mejor opción para la evaluación de productos matriz-vector en varias dimensiones. Sin embargo, nuestras pruebas numéricas sugieren que hay una compensación mutua entre el tamaño del parámetro de forma de la función multicuádrica y el número de puntos en la discretización espacial del problema de valores iniciales. Modificar la estrategia descrita a lo largo de esta tesis para sobreponerse a esta problemática queda como un problema abierto. Es oportuno mencionar que problemáticas similares aparecen en otros contextos de interpolación con funciones de base radial y se les considera un tema de investigación.

# Indice

Dedicatoria	iii
Agradecimientos	iv
Resumen	v
<b>1 Introducción</b>	<b>1</b>
<b>2 Planteamiento del problema</b>	<b>5</b>
2.1 Opciones financieras . . . . .	6
2.2 Modelo de Black-Scholes . . . . .	8
2.3 Problema de Valores Iniciales de Black-Scholes (PVIBS) . . . . .	12
2.3.1 Existencia y unicidad de soluciones del PVIBS . . . . .	13
2.3.2 Fórmula de Black-Scholes . . . . .	14
2.4 Opciones para una canasta de activos . . . . .	16
<b>3 Solución Numérica del PVIBS</b>	<b>18</b>
3.1 Funciones de Base Radial (FBR) . . . . .	19
3.2 Solución del PVIBS en un dominio acotado . . . . .	23
3.3 Solución del PVIBS por medio de FBR . . . . .	23
3.4 Discretización temporal . . . . .	24

3.5	Costo computacional . . . . .	27
<b>4</b>	<b>Aceleración de los cálculos</b>	<b>29</b>
4.1	Métodos Iterativos . . . . .	30
4.1.1	Precondicionamiento con funciones cardinales aproximadas . . . . .	31
4.1.2	El algoritmo iterativo . . . . .	33
4.2	Multiplicación Rápida de Matrices . . . . .	36
4.2.1	Suma rápida de Multicuádricas con el Método Multipolos	37
4.2.2	Suma Rápida de Multicuádricas usando la Transformada Rápida de Gauss . . . . .	44
<b>5</b>	<b>Resultados</b>	<b>56</b>
5.1	Solución del PVIBS usando métodos iterativos . . . . .	57
5.2	Suma rápida de multicuádricas con el Método Multipolos . . . . .	62
5.3	Solución del PVIBS utilizando el Método Multipolos . . . . .	64
5.4	Suma rápida de multicuádricas basada en la Transformada Rápida de Gauss Mejorada . . . . .	65
<b>6</b>	<b>Conclusiones</b>	<b>71</b>
	<b>Bibliografía</b>	<b>73</b>



# Capítulo 1

## Introducción

Las matemáticas financieras se han convertido en un área de investigación muy activa desde el punto de vista del análisis numérico a partir de la aparición de los trabajos de Merton [27] y Black y Scholes [9].

Se ha establecido que el problema de asignar un precio a las opciones financieras se puede formular como un problema de Cauchy para una ecuación diferencial parcial parabólica (en lo sucesivo nos referiremos a este como el problema de valores iniciales de Black-Scholes ó PVIBS). La importancia de desarrollar métodos numéricos para resolver eficientemente el PVIBS es evidente.

En este trabajo proponemos aproximar la solución del problema que nos ocupa mediante interpolación con funciones de base radial. Para esto usamos herramientas estándar tales como algoritmos iterativos para interpolación con funciones de base radial [22] y métodos de sumación rápida como el método multipolos [20], [6], o los métodos basados en la Transformada Rápida de Gauss [18]. Nuestra motivación para estudiar la solución numérica del PVIBS mediante interpolación con funciones de base radial consta de dos partes. Primeramente, la rápida convergencia de la solución mediante interpolación con funciones de base

radial de problemas de valores a la frontera para problemas elípticos lineales, sugiere que es razonable esperar resultados análogos para el PVIBS. Por otra parte, aunque los sistemas de ecuaciones lineales asociados al problema de interpolación son densos, las matrices correspondientes típicamente tienen pocos grados de libertad, es decir, el problema de interpolación es intrínsecamente de bajo costo computacional.

El contenido de este trabajo se distribuye como sigue:

- **Capítulo 2.** En este capítulo se presenta el problema de asignar precio a las opciones financieras. Se describe brevemente el modelo de Black-Scholes [3], y cómo es que de este se obtiene una ecuación diferencial parcial que permite calcular el precio de las opciones.
- **Capítulo 3.** A lo largo de este capítulo se resume el uso de las funciones de base radial en la interpolación de datos [10, 16] y su aplicación a la solución del Problema de Valores Iniciales de Black-Scholes [12, 26]. Además se muestran algunos métodos comunmente empleados para la marcha en el tiempo de este tipo de problema.
- **Capítulo 4.** En este capítulo se presenta el método iterativo [22] utilizado en este trabajo para resolver el problema de interpolación que surge al avanzar en el tiempo. También se resume el uso de los métodos Multipolos [32] y basados en Transformada Rápida de Gauss [18] para la evaluación rápida de multicuádricas.
- **Capítulo 5.** Aquí se muestran los resultados obtenidos en el trabajo de tesis. El contenido de este capítulo está organizado como sigue:

- En la sección 5.1 se detalla cómo aplicar el método iterativo descrito en [22] para resolver los sistemas de ecuaciones que surgen al marchar en el tiempo.
- En la sección 5.2 se presentan algunos detalles de nuestra implementación del método multipolos, es decir, cuáles son los operadores de traslación de expansiones utilizados en este trabajo. También se muestran algunas pruebas de la eficiencia de evaluación de multicuádricas utilizando este método.
- En la sección 5.3 comparan los tiempos de ejecución de las tres formas de aproximar la solución del PVIBS utilizadas en este trabajo.

**1** Invirtiendo la matriz de interpolación.

**2** Utilizando el algoritmo iterativo con multiplicación directa de matrices.

**3** Utilizando el algoritmo iterativo con el método multipolos como evaluador de productos matriz vector.

- En la sección 5.4 se presenta una fórmula similar a la mostrada en [18], que permite evaluar funciones de base radial multicuádricas inversas generalizadas (exponente negativo arbitrario), y se presentan algunas pruebas de eficiencia utilizando la Transformada Rápida de Gauss Mejorada [11] de Duraiswami *et al* para la evaluación funciones de base radial Gaussianas.

- **6 Conclusiones**

Los resultados obtenidos en este trabajo se pueden resumir como sigue:

- Las pruebas realizadas proporcionan evidencia numérica de que al aplicar

métodos iterativos en la solución del PVIBS, se obtiene un ahorro sustancial en el tiempo requerido para resolver las ecuaciones de interpolación que surgen al avanzar en el tiempo.

- Al combinar un algoritmo iterativo para resolver sistemas de ecuaciones lineales con un método rápido para la evaluación de productos matriz vector, se aceleran de manera importante los cálculos requeridos en la solución de PVIBS.
- Para aproximar la solución del PVIBS en dimensiones mayores a uno, se considera el método rápido para la evaluación de multicuádras propuesto en [18] en combinación con la Transformada Rápida de Gauss Mejorada (eficiente en altas dimensiones), como un buen candidato para aplicarlo en la solución de este problema. Después de realizar varias pruebas numéricas, se identificó que este método es sensible al parámetro de forma. Queda abierto el problema de mejorar este método de evaluación de FBR multicuádras para valores pequeños del parámetro de forma.

# Capítulo 2

## Planteamiento del problema

El modelo Black-Scholes está basado en una ecuación diferencial estocástica, de la cual, mediante el cálculo de  $\hat{\text{Ito}}$ , se puede derivar en una ecuación diferencial parcial (EDP) [13] que describe el comportamiento del precio de las opciones financieras. Por tanto, el problema de asignar valores a los derivados financieros equivale, bajo hipótesis estándar, a resolver un problema de valores iniciales para una EDP parabólica. Existen numerosos esfuerzos por desarrollar métodos numéricos eficientes para resolver el PVIBS. Una referencia panorámica sobre métodos numéricos para dicho problema es el libro de Achdou y Pironneau [3].

Con el propósito de que este trabajo sea autocontenido, en este capítulo ofrecemos una derivación del PVIBS. El contenido del capítulo se distribuye como sigue: en la sección 2.1 ofrecemos algunas definiciones generales. En la sección 2.2 presentamos la definición del modelo de Black-Scholes. En la sección 2.3 se ofrece una deducción estándar de el PVIBS, junto con resultados de existencia y unicidad de la solución. La discusión está basada en [3], [13] y [35].

## 2.1 Opciones financieras

Una opción es un contrato que da a su propietario el derecho, pero no la obligación, de comprar o vender bienes o valores, a un precio determinado, en una fecha concreta.

Al comprar una opción se debe especificar:

- El tipo de opción: *call* para la compra, y *put* para la venta.
- El tipo de bien a tratar (el activo subyacente): acciones, índices bursátiles, etc.
- La cantidad del activo subyacente a ser comprado o vendido.
- La fecha de vencimiento o madurez ( $T$ ).
- El precio de ejercicio ( $K$ ), que es el precio al cual se hace la transacción si la opción es ejercida.

Si la opción puede ser ejercida en cualquier tiempo hasta la madurez, entonces se le llama opción Americana. Si sólo puede ser ejercida en la fecha de vencimiento, entonces se conoce como opción Europea. En lo siguiente sólo se hablará de opciones Europeas, que son las más simples.

**Ejemplo 2.1.1** *Consideremos una opción call para la compra de una acción de Microsoft. Esta le da a su titular el derecho a comprar una acción (cantidad del activo) por la cantidad de \$25 ( $K = 25$ ) dentro de un mes ( $T = 1$ ).*

*Si el precio de dicha acción para el día de hoy es \$24.5, y permanece igual hasta la fecha de vencimiento. ¿El titular debería ejercer la opción? La respuesta es no, dado que tendría que comprar una acción de Microsoft por \$25, mientras que en el mercado sólo vale \$24.5.*

*¿Pero qué pasa si el precio de la acción sube a \$29? Ahora, en caso de ejercer, obtendría una ganancia de  $\$29 - \$25 = \$4$ .*

Considerando la idea presentada en el ejemplo 2.1.1, si denotamos por  $S_T$  el precio del activo en la fecha de vencimiento, se observa que:

- Si  $S_T < K$  titular no debe ejercer la opción.
- En cambio, si  $S_T > K$ , el titular obtiene una ganancia de  $S_T - K$ , en caso de ejercer.

Entonces el valor de una opción en la fecha de vencimiento está dado por la ganancia, en caso de ejercer, y es nulo en otro caso.

El valor de una opción call en la fecha de vencimiento está dado por:

$$(S_T - K)_+ = \max(S_T - K, 0). \quad (2.1)$$

Por otro lado, el titular de una opción put quiere que el precio del activo caiga, así podría venderlo en el mercado por más de lo que vale, es decir, el valor de una opción put en la fecha de vencimiento está dado por:

$$(K - S_T)_+ = \max(K - S_T, 0). \quad (2.2)$$

Las funciones (2.1) y (2.2) son conocidas como función de pago para una opción call y put respectivamente. De la función de pago se puede deducir que a mayor precio de ejercicio, menor es el valor de una opción call, y el valor de una opción put es menor cuando el precio de ejercicio disminuye, en la fecha de vencimiento. Esto se puede apreciar mejor en la figura 2.1.

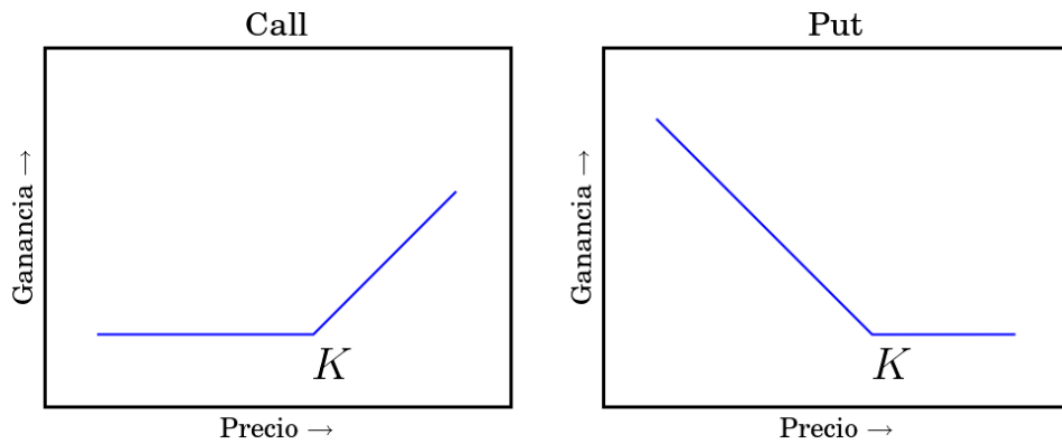


Figura 2.1: Función de pago ( $t = T$ ).

Hasta este punto, se conoce el valor de una opción en la fecha de vencimiento, pero una pregunta importante que se puede hacer es ¿Cuánto vale la opción ahora?, es decir, ¿Cuánto se debe pagar por un contrato que otorga ciertos derechos en el futuro?

## 2.2 Modelo de Black-Scholes

Bajo ciertas hipótesis sobre el mercado, el modelo sugerido por Black y Scholes describe el comportamiento del precio del activo subyacente. Estas hipótesis son:

- Las transacciones no tienen costo y son instantáneas.
- Se supone que no existe arbitraje, es decir, en ningún momento es posible obtener un beneficio sin asumir riesgo alguno.

El modelo de Black-Scholes considera un activo con riesgo (el activo subyacente) con precio  $S_t$  al tiempo  $t$ , y un activo sin riesgo con precio  $S_t^0$  al tiempo  $t$ ,



además, para describir el comportamiento de  $S_t$ , supone que este satisface una Ecuación Diferencial Estocástica de tiempo continuo.

También se supone que el precio del activo sin riesgo satisface la siguiente ecuación diferencial ordinaria

$$dS_t^0 = r(t)S_t^0 dt, \quad (2.3)$$

donde  $r(t)$  es una tasa de interés libre de riesgo.

Si se pone  $S_0^0 = 1$  en la ecuación (2.3), se obtiene que:

$$S_t^0 = \exp\left(\int_0^t r(\tau) d\tau\right), \quad S_t^0 = \exp(rt) \text{ si } r \text{ es constante.}$$

Para describir el precio del activo, en el modelo de Black-Scholes descompone el rendimiento  $\frac{dS_t}{S_t}$  como la suma de un término determinista  $\mu dt$ , y un término aleatorio  $\sigma_t dB_t$ , en el cuál se modelan las variaciones del precio debido a causas externas.

De manera más precisa, en el modelo de Black-Scholes se supone que el precio del activo con riesgo es una solución de la siguiente Ecuación Diferencial Estocástica

$$dS_t = S_t(\mu dt + \sigma_t dB_t), \quad t \in [0, T] \quad (2.4)$$

donde  $B_t$  es un movimiento Browniano estándar en un espacio de probabilidad  $(\Sigma, \mathcal{A}, \mathbb{P})$ ,  $\sigma_t$  es la volatilidad (una medida de la incertidumbre en el precio) y  $\mu$  el drift (una tasa de crecimiento promedio del precio).

El significado preciso de (2.4) se puede consultar en [25, 13].

Si  $\sigma_t = \sigma$  es una constante, se puede deducir que la solución de la ecuación (2.4) se puede escribir en términos de  $B_t$

$$S_t = S_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma B_t\right).$$

Hasta este punto se cuenta con una Ecuación Diferencial Estocástica que describe el comportamiento del precio de un activo, pero, ¿En qué sentido es útil la ecuación (2.4) para calcular el precio a la opción?

Del modelo de Black-Scholes se puede deducir una fórmula para asignar precio a la opción en un tiempo  $t < T$  en términos del valor esperado de una variable aleatoria. El siguiente argumento puede ser encontrado en [13].

Por el teorema de Girsanov, existe una probabilidad  $\mathbb{P}^*$  equivalente a  $\mathbb{P}$  (la medida del “mundo real”) tal que el precio  $S_t$  satisface

$$dS_t = S_t(rdt + \sigma_t dW_t) \quad (2.5)$$

donde

$$W_t = B_t + \int_0^t \frac{\mu - r}{\sigma}$$

es un movimiento browniano estándar bajo  $\mathbb{P}^*$ , es decir, bajo la probabilidad  $\mathbb{P}^*$  el precio descontado  $\tilde{S}_t = S_t e^{-rt}$  satisface

$$d\tilde{S}_t = \sigma \tilde{S}_t dW_t,$$

por lo cual es una martingala, ver [13], [25].

Se puede probar que es posible simular la opción por medio de un portafolio autofinanciado que contiene las cantidades  $H_t$ ,  $H_t^0$ , del activo con riesgo y sin riesgo respectivamente, es decir, existe un par de procesos adaptados tales que

$$\begin{aligned} \int_0^T |H_t^0| dt + \int_0^T H_t^2 dt &< \infty \quad \text{casi en todas partes (c.t.p.)}, \\ H_t^0 S_t^0 + H_t S_t &= H_0^0 S_0^0 + H_0 S_0 + \int_0^t H_\tau^0 dS_\tau^0 \\ &+ \int_0^t H_\tau dS_\tau \quad \text{c.t.p. } \forall t \in [0, T], \quad \text{y} \\ C_T &= H_T^0 S_T^0 + H_T S_T. \end{aligned}$$

Por la hipótesis de no arbitraje, el precio de la opción debe estar dado por

$$C_t = H_t^0 S_t^0 + H_t S_t. \quad (2.6)$$

También es posible ver que para un portafolio auto financiado se cumple que

$$\begin{aligned} H_t^0 + H_t \tilde{S}_t &= H_0^0 + H_t \tilde{S}_t^0 + \int_0^t H_\tau d\tilde{S}_\tau. \quad \text{c.t.p. } \forall t \in [0, T], \\ &= H_0^0 + H_t \tilde{S}_t^0 + \int_0^t \sigma H_\tau d\tilde{W}_\tau, \end{aligned}$$

lo cuál implica que  $H_t^0 + H_t \tilde{S}_t$  es una martingala cuadrado integrable bajo  $\mathbb{P}^*$ .

Tomando en cuenta (2.6), el precio de la opción está dado por

$$C_t = e^r \mathbb{E}^* [(C_T e^{-rT} | F_t)] = \mathbb{E}^* [C_T e^{-r(T-t)} (S_T - K)_+ | F_t] \quad (2.7)$$

donde  $F_t$  es la filtración natural asociada a  $W_t$ .

**Nota 2.2.1** *Los resultados mencionados en esta sección se pueden generalizar al caso en que  $\sigma_t = \sigma(S_t, t)$  y  $r_t = r(t)$ , con  $r$  y  $\sigma$  funciones continuas tales que  $S\sigma(S, t)$  es Lipschitz con respecto a  $S$  con una constante de Lipschitz independiente de  $t$ ,  $\sigma$  acotada por arriba, y acotada por abajo por una constante mayor que cero, uniformemente en  $t$ .*

En dicho caso el precio de la opción está dada por

$$C_t = \mathbb{E}^* \left[ \exp \left( - \int_0^t r(\tau) d\tau \right) P_0(S_T) | F_t \right].$$

donde  $P_0(S)$  es la función de pago .

**Nota 2.2.2** *En el modelo de Black-Scholes con coeficientes uniformes, comúnmente es difícil recuperar el precio de las opciones en el mercado, así que una práctica común es calibrar una función de volatilidad  $\sigma(S, t)$  que se ajuste a los precios del mercado.*

## 2.3 Problema de Valores Iniciales de Black-Scholes

En la sección anterior se vio que el precio de una opción está dado por la fórmula

$$\phi(S_t, t) = \mathbb{E}^* \left[ \exp \left( \int_0^t r(\tau) d\tau \right) P_0(S_T) | F_t \right], \quad (2.8)$$

es decir, el Modelo de Black-Scholes permite calcular el precio de una opción como el valor esperado de una variable aleatoria. Con ayuda del cálculo de Itô es posible deducir una EDP que describe dicho precio, es decir, es posible relacionar la función  $\phi$  en (2.8) con la solución de una EDP.

La proposición 2.3.1 y el teorema 2.3.1 establecen esta relación.

**Proposición 2.3.1** *Sean  $\sigma, r$  funciones continuas, no negativas y acotadas, en  $\mathbb{R}_+ \times [0, T]$  y  $[0, T]$  respectivamente, y que  $S \mapsto S\sigma(S, t)$  es Lipschitz con una constante independiente de  $t$ . Entonces, para cualquier función  $u : (S, t) \mapsto u(S, t)$  continua en  $\mathbb{R} \times [0, T)$ , y tal que  $|S \frac{\partial u}{\partial S}| \leq C(1 + S)$  con  $C$  independiente de  $t$ , el proceso*

$$M_t = \exp \left( - \int_0^t r(\tau) d\tau \right) u(S_t, t) - \int_0^t \exp \left( - \int_0^\tau r(v) dv \right) \left( \frac{\partial u}{\partial t}(S_t, \tau) + L_\tau u(S_\tau, \tau) - r(\tau)u(S_\tau, \tau) \right) d\tau$$

es una martingala bajo  $F_t$ , dónde  $L_t$  es el operador diferencial dado por:

$$L_t f(S) = \frac{\sigma^2(S, t) S^2}{2} \frac{d^2}{dS^2} f(S) + r(t) S \frac{d}{dS} f(S).$$

El operador  $L_t$  se llama el generador infinitesimal del proceso de Markov  $S_t$ .

**Teorema 2.3.1** *Sean  $\sigma, r$  funciones continuas, no negativas y acotadas, y  $S \mapsto S\sigma(S, t)$  Lipschitz con una constante independiente de  $t$ . Consideremos una*

función  $P : \mathbb{R}_+ \times [0, T] \rightarrow \mathbb{R}$ , continua en  $\mathbb{R}_+ \times [0, T]$ ,  $\mathcal{C}^1$  con respecto de  $t$  y  $\mathcal{C}^2$  con respecto a  $S$  en  $\mathbb{R}_+ \times [0, T)$ , tal que  $|S \frac{\partial u}{\partial S}| \leq C(1 + S)$  con  $C$  independiente de  $t$ , y supóngase que  $P$  satisface

$$\frac{\partial}{\partial t} P(S, t) = rP(S, t) - rS \frac{\partial}{\partial S} P(S, t) - \frac{\sigma^2 S^2}{2} \frac{\partial^2}{\partial S^2} P(S, t), \quad (2.9)$$

$$P(S, T) = P_T(S), \quad S \in \mathbb{R}_+ \quad (2.10)$$

entonces, para  $\phi$  dada por (2.8) se cumple que  $\phi = P$ .

La ecuación (2.9) es una EDP parabólica, y el problema conformado por (2.9), (2.10) es un problema de Cauchy hacia atrás en el tiempo, al cuál nos referimos como Problema de Valores Iniciales de Black-Scholes (PVIBS).

### 2.3.1 Existencia y unicidad de soluciones del PVIBS

El PVIBS (2.9), (2.10) es un problema clásico y la regularidad de sus soluciones ha sido bastante estudiada. Se puede consultar [17] para ver detalles sobre lo enunciado a continuación.

**Proposición 2.3.2** Sea  $\Omega = \mathbb{R}^+ \times [0, T]$ , si:

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

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

### 2.3.2 Fórmula de Black-Scholes

Cuando los parámetros  $\sigma(S, t)$  y  $r(t)$  son constantes, el PVIBS tiene una solución conocida, llamada Fórmula de Black-Scholes.

Una manera de derivar esta fórmula, es utilizar la solución fundamental de la ecuación de difusión.

Por ejemplo, consideremos una opción call, el Problema correspondiente es:

$$\frac{\partial}{\partial t} C(S, t) = rC(S, t) - rS \frac{\partial}{\partial S} C(S, t) - \frac{\sigma^2 S^2}{2} \frac{\partial^2}{\partial S^2} C(S, t) \quad (2.11)$$

$$C(S, T) = \max(S - K, 0) \quad (2.12)$$

Primero se hace un cambio de variable, que permite transformar (2.11) en una EDP con coeficientes constantes

$$\theta = T - t, \quad x = \ln S, \quad \phi(x, \theta) = C(e^x, T - \theta),$$

calculando las derivadas de  $C$ , se obtiene que

$$S \frac{\partial C}{\partial S} = \frac{\partial \phi}{\partial x}, \quad S^2 \frac{\partial^2 C}{\partial S^2} = \frac{\partial^2 \phi}{\partial x^2} - \frac{\partial \phi}{\partial x},$$

entonces el PVIBS (2.11),(2.12) se transforma en

$$\frac{\partial \phi}{\partial \theta} = \frac{1}{2} \sigma^2 \frac{\partial^2 \phi}{\partial x^2} + \left( r - \frac{\sigma^2}{2} \right) \frac{\partial \phi}{\partial x} - r\phi \text{ en } \mathbb{R} \times (0, T] \quad (2.13)$$

$$\phi(x, 0) = \max(e^x - K, 0),$$

$$\phi(x, \theta) \rightarrow 0 \text{ cuando } x \rightarrow -\infty, \quad (2.14)$$

$$\phi(x, \theta) = e^x - Ke^{-r\theta} \text{ cuando } x \rightarrow +\infty. \quad (2.15)$$

Ahora, se realiza un cambio de variable en la variable dependiente, lo cual transforma la ecuación (2.13) en la ecuación de difusión.

Sea

$$\begin{aligned}\phi(x, \theta) &= \varphi(x, \theta)e^{a\theta+bx}, \quad \text{dónde} \\ b &= \frac{1}{2} - \frac{r}{\sigma^2},\end{aligned}\tag{2.16}$$

$$a = -r - \sigma^2 \frac{b^2}{2},\tag{2.17}$$

entonces  $\phi$  satisface la ecuación

$$\begin{aligned}\frac{\partial \varphi}{\partial \theta} - \frac{1}{2}\sigma^2 \frac{\partial^2 \varphi}{\partial x^2} &= 0 \quad \text{en} \quad \mathbb{R} \times (0, T], \\ \varphi(x, 0) &= \max \left( e^{(1/2+r/\sigma^2)x} - Ke^{(-1/2+r/\sigma^2)x}, 0 \right).\end{aligned}$$

La solución de este problema de valores iniciales es conocida:

$$\varphi(x, \theta) = \frac{1}{2\pi\sigma^2\theta} \int_{\mathbb{R}} \exp\left(-\frac{y^2}{2\sigma^2\theta}\right) \varphi(x-y, 0) dy.$$

Si se invierte el cambio de variable realizado, se obtiene la Fórmula de Black-Scholes para una opción call:

$$C(S, t) = SN(d_1) - Ke^{-r(T-t)}N(d_2)\tag{2.18}$$

dónde  $d_1, d_2$  y  $N$  en (2.18) y (2.13) están dados por:

$$\begin{aligned}d_1 &= \frac{\ln \frac{S}{K} + \left(r + \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}} \\ d_2 &= d_1 - \sigma\sqrt{T-t} \\ N(d) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^d e^{-\frac{x^2}{2}} dx.\end{aligned}$$

Mediante un proceso similar se puede derivar la Fórmula de Black-Scholes correspondiente a una opción put:

$$C(S, t) = -SN(-d_1) + Ke^{-r(T-t)}N(-d_2).\tag{2.19}$$

## 2.4 Opciones para una canasta de activos

Existen varios productos financieros contruidos a partir una canasta de activos, lo cual resulta en problemas en dimensión  $d$ , donde  $d$  es el número de activos. Asignar precio a este tipo de opciones problema representa un gran reto computacionalmente, por lo que la solución numérica del problema del PVIBS para varias dimensiones es un área de investigación muy activa.

El problema de asignar precio a las opciones correspondientes a una canasta de activos se puede modelar de manera similar al caso unidimensional presentado en las secciones anteriores, salvo que es necesario introducir ciertos factores de correlación entre los activos correspondientes.

Consideremos una canasta con  $d$  activos, cuyos precios  $S_{it}, i = 1, \dots, d$  satisfacen:

$$dS_{it} = S_{it} (\mu_i dt + \sigma_i dB_{it}), \quad (2.20)$$

dónde  $B_{1t}, \dots, B_{dt}$  son  $d$  movimientos brownianos estándar bajo la probabilidad  $\mathbb{P}$  y  $\mu_i, \sigma_i$  son el drift y volatilidad correspondientes. Ahora los precios de los activos están correlacionados, es decir,

$$\mathbb{E}[S_{it}S_{jt}] = \rho_{ij},$$

dónde la matriz  $\Psi = [\rho_{i,j}]$  se conoce como la matriz de correlación, que es simétrica, positivo definida, y  $\rho_{ii} = 1, i = 1, 2, \dots, d$ . La matriz de covarianza está dada por:

$$\Xi = \Sigma\Psi\Sigma$$

dónde  $\Sigma = \text{diag}(\sigma_i)$  es la matriz de volatilidad.

La opción para esta canasta de activos con madurez  $T$  y para una función  $Q^0 : \mathbb{R}_+^d \rightarrow \mathbb{R}_+$  puede ser ejercida en tiempo  $t = T$  para un pago  $Q^0(S_{1T}, \dots, S_{dT})$ .



Es posible encontrar una probabilidad libre de riesgo  $\mathbb{P}^*$  bajo la cual el precio de la opción está dado por:

$$Q_t = \mathbb{E}^* \left( \exp \left( \int_t^T -r(\tau) d\tau \right) Q^0(S_{1T}, \dots, S_{dT}) | F_r \right). \quad (2.21)$$

La fórmula (2.21) permite calcular el precio de la opción correspondiente como el valor esperado de una variable aleatoria, pero también es posible deducir una EDP que describe el comportamiento de  $Q$ :

$$\begin{aligned} \frac{\partial Q}{\partial t}(S, t) + \mathcal{L}Q(S, t) &= 0, t < T, S \in \mathbb{R}_+^d, \\ Q(S, T) &= Q^0(S), S \in \mathbb{R}_+^d \end{aligned} \quad (2.22)$$

dónde (2.22) es la ecuación de Black-Scholes multidimensional con el operador espacial  $\mathcal{L}$  dado por:

$$\mathcal{L}Q(S, t) = r \sum_{i=1}^d s_i \frac{\partial Q}{\partial s_i} + \frac{1}{2} \sum_{i,j=1}^d \Xi_{ij} s_i s_j \frac{\partial^2 Q}{\partial s_i \partial s_j} - rQ. \quad (2.23)$$

La función de pago  $Q$  puede definirse de distintas maneras, por ejemplo, entre las más populares se encuentran

$$\begin{aligned} Q^0 &= \max \left( \frac{1}{d} \sum_{i=1}^d s_i - K, 0 \right) \\ Q^0 &= \max \left( \sum_{i=1}^d s_i - K, 0 \right) \\ Q^0 &= \max (\max(s_i) - K, 0). \end{aligned}$$

# Capítulo 3

## Solución Numérica del PVIBS

Existen varios métodos numéricos estándar para resolver problemas de valores iniciales como el PVIBS (2.9, 2.10), por ejemplo:

- diferencias finitas,
- elementos finitos,
- volúmenes finitos,

entre otros.

En 1971, el geofísico Rolland Hardy [23] introdujo un método para la interpolación de datos geográficos. Resultados numéricos mostraron que las aproximaciones obtenidas con este método eran bastante precisas. Posteriormente, el físico Ed Kansa [15], propuso un método de colocación para la solución de EDPs basado en la técnica de interpolación descrita por Hardy.

La aparición de estos trabajos dio lugar al método de las Funciones de Base Radial (FBR), el cual ha sido aplicado desde entonces de manera exitosa en muchas áreas.

Una ventaja del uso FBR es que proveen métodos numéricos libres de malla para la solución de EDPs, y además la solución de problemas en altas dimensiones es bastante similar al caso unidimensional. Por otro lado, los sistemas de ecuaciones resultantes son generalmente densos (a excepción de las FBR de soporte compacto), sin embargo, las matrices correspondientes tienen pocos grados de libertad.

Otra ventaja al seguir este enfoque en el contexto del PVIBS, es que se pueden elegir FBR continuamente derivables, y las derivadas parciales de la solución se pueden obtener directamente calculando las derivadas de las FBR, lo cual es útil para calcular derivados financieros como los valores Delta.

El contenido de este capítulo se distribuye como sigue: en la sección 3.1 se introduce el uso de funciones de base radial para la interpolación de datos. En la sección 3.2 se discuten algunos detalles sobre la solución numérica del PVIBS en un dominio acotado. En las secciones 3.3 y 3.4 se ilustra el uso de FBR en la solución numérica del PVIBS, así como esquemas estándar para la discretización en el tiempo de la Ecuación Diferencial Ordinaria que surge en este problema. Por último, en la sección 3.5 se discute el costo computacional para los métodos discutidos en este capítulo.

## 3.1 Funciones de Base Radial

Aunque la teoría general de funciones de base radial no ha sido completamente establecida, trabajos tales como [16, 10] ofrecen buen material introductorio sobre interpolación de datos y solución de Ecuaciones Diferenciales Parciales por medio de FBR.

El problema de interpolación de datos se puede establecer como sigue:

**Definición 3.1.1** *Dados los datos  $(x_j, y_j), j = 1, \dots, N$  con  $x_j \in \mathbb{R}^d$ , encontrar una función  $s$ , tal que  $s(x_j) = y_j, j = 1, \dots, N$ .*

Los  $x_j$  son los puntos donde se tomaron medidas y los  $y_j$  son las medidas correspondientes.

Una forma común de resolver el problema de interpolación de datos 3.1.1 es suponiendo que  $s$  es una combinación lineal de ciertas funciones base  $B_k$ , es decir,

$$s(x) = \sum_{k=1}^N \lambda_k B_k(x), x \in \mathbb{R}^d, \quad (3.1)$$

entonces, bajo esta hipótesis, resolver el problema 3.1.1 da lugar a un sistema de ecuaciones lineales de la forma

$$\mathbf{A}\lambda = y \quad (3.2)$$

donde  $A_{jk} = B_k(x_j), c = [c_j]$  y  $y = [y_j], j, k = 1, \dots, N$ .

**Definición 3.1.2** *Una función  $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}$  se llama función de base radial si existe una función  $\phi : [0, \infty) \rightarrow \mathbb{R}$  tal que*

$$\Phi(x) = \phi(r), \text{ donde } r = \|x\|,$$

y  $\|\cdot\|$  es alguna norma en  $\mathbb{R}^d$  (normalmente la norma Euclidiana).

Si usamos FBR como funciones base, es decir  $B_k(x) = \Phi(x - x_k)$  entonces la función de interpolación (3.1) es

$$s(x) = \sum_{k=1}^N \lambda_k \Phi(x - x_k), x \in \mathbb{R}^d. \quad (3.3)$$

**Nota 3.1.1** *Lo que hace que una FBR sea útil en las aplicaciones, es que los problemas de interpolación que se derivan no son sensibles a la dimensión de los datos.*

En lo sucesivo referiremos a FBR indistintamente  $\Phi$ , y a la función de una variable  $\phi$  correspondiente.

**Definición 3.1.3** *Una matriz real  $A$  es semi-definida positiva si*

$$\sum_{j=1}^N \sum_{k=1}^N c_j c_k A_{jk} \geq 0 \quad (3.4)$$

para  $c = [c_j] \in \mathbb{R}^N$ . Si el único valor de  $c$  para el cual (3.4) se convierte en igualdad es  $c = 0$ , entonces  $A$  se llama definida positiva.

Una propiedad importante de las matrices definidas positivas es que todos sus valores propios son positivos, es decir, una matriz positivo definida es no singular. Entonces, si se tiene FBR en (3.3) que generan una matriz de interpolación definida positiva, el problema de interpolación tiene solución. Existen varias FBR estándar que generan matrices definidas positivas.

Nombre	$\phi(r)$	parámetro
<i>Gaussiana</i>	$e^{-c^2 r^2}$	
<i>Multicuádrica Inversa</i>	$(r^2 + c^2)^{-k/2}$	$k > 0$ .
<i>Sobolev</i>	$K_\nu(r) r^\nu$	$\nu > 0$ .

Hay FBR útiles en las aplicaciones que no generan matrices definidas positivas. En tales casos, se debe agregar un polinomio de cierto grado a la función de interpolación (3.3).

Sea  $P_{Q-1}^d$  el espacio generado por todos los polinomios en  $d$  variables, de grado a lo más  $Q - 1$ , y  $p_1, \dots, p_q$  una base. Si se agrega un polinomio del espacio  $P_{Q-1}^d$  a  $s$ , se obtiene

$$s(x) = \sum_{k=1}^N \lambda_k \Phi(x - x_k) + \sum_{l=0}^q \beta_l p_l(x), \quad (3.5)$$

donde  $q = \binom{Q-1+d}{d}$  es la dimensión de  $P_{Q-1}^d$ .

Ahora se tienen  $q$  grados de libertad adicionales, entonces, para completar el sistema se agregan  $q$  ecuaciones homogéneas

$$\sum_{k=1}^N \lambda_k p_l(x_k) = 0, 1 \leq l \leq q. \quad (3.6)$$

El sistema de ecuaciones resultante para este problema de interpolación es

$$\begin{aligned} s(x) = \sum_{k=1}^N \lambda_k \Phi(x - x_k) + \sum_{l=0}^q \beta_l p_l(x) &= y_j, 1 \leq j \leq N \\ \sum_{k=1}^N \lambda_k p_l(x_k) &= 0, 1 \leq l \leq q, \end{aligned} \quad (3.7)$$

cuya solución se asegura si  $p \in P_{Q-1}^d$  y  $p(x_k) = 0, 1 \leq k \leq N$  implica que  $p = 0$ .

La tabla 3.1 muestra algunas de las FBR condicionalmente positivo definidas más comunes.

Nombre	$\phi(r)$	parámetro	$Q$
<i>Multicuádrica</i>	$(-1)^{\lceil k/2 \rceil} (r^2 + c^2)^{k/2}$	$k > 0, k \notin 2\mathbb{Z}$	$\lceil k/2 \rceil$
<i>Poliarmónica</i>	$(-1)^{\lceil k/2 \rceil} r^k$	$k > 0, k \notin 2\mathbb{Z}$	$\lceil k/2 \rceil$
<i>Poliarmónica</i>	$(-1)^{k/2+1} r^k \log r$	$k > 0, k \in 2\mathbb{Z}$	$k/2 + 1$
<i>Placa delgada</i>	$r^2 \log r$		2

## 3.2 Solución del PVIBS en un dominio acotado

El principio del máximo [3] es una herramienta para obtener estimaciones de la solución de EDPs elípticas y parabólicas, en particular nos asegura que se puede aproximar la solución del PVIBS (2.9), (2.10) por la solución de un problema en un dominio espacial acotado.

**Proposición 3.2.1** *Supóngase que existen constantes positivas  $\underline{\sigma}, \bar{\sigma}$  y  $C_\sigma$ , tales que para todo  $t \in [0, T]$  y todo  $S \in \mathbb{R}_+$ , se cumple que*

- $0 \leq \underline{\sigma} \leq \sigma(S, t) \leq \bar{\sigma}$ ,
- $|S \frac{\partial \sigma}{\partial S}(S, t)| \leq C_\sigma$ ,

entonces el error  $\max_{t \in [0, T], S \in [0, \tilde{S}]} |P(S, t) - \tilde{P}(S, t)|$  decae más rápido que cualquier exponencial  $\exp(-\eta \tilde{S})$ , ( $\eta > 0$ ) cuando  $S \rightarrow \infty$ .

donde  $P(S, t), \tilde{P}(S, t)$  es la solución del PVIBS en  $\mathbb{R}_+$  y  $[0, \tilde{S}]$ , respectivamente.

Para poder implementar apropiadamente la solución numérica del PVIBS sobre un dominio espacial acotado, se deben imponer condiciones de frontera artificiales, a partir de las condiciones asintóticas (2.14),(2.15)

$$V(0, t) = Ke^{-r(T-t)}, \quad \lim_{S \rightarrow \infty} V(S, t) = 0, \quad \text{put} \quad (3.8)$$

$$V(0, t) = 0, \quad \lim_{S \rightarrow \infty} V(S, t) = S, \quad \text{call.} \quad (3.9)$$

## 3.3 Solución del PVIBS por medio de FBR

Existen varios trabajos en los que se ha hecho uso de FBR para aproximar la solución de PVIBS [12, 29, 26]. En estos trabajos se presenta evidencia numérica sobre la eficiencia de este método.

Considérese el PVIBS (2.9),(2.10) en un dominio espacial truncado, con datos finales  $V(s_i, T) = V_T(s_i)$ ,  $s_i \in [0, \tilde{S}]$ ,  $i = 1, \dots, N$ ,  $\tilde{S} > 0$ .

Se aproxima  $V(s, t)$  como una suma de FBR, con coeficientes  $\alpha_j(t)$

$$V(s, t) \approx v(s, t) = \sum_{j=1}^N \alpha_j(t) \Phi(s - s_j). \quad (3.10)$$

Sustituyendo  $v(s, t)$  en el PVI, y evaluando en  $s_i, i = 1, \dots, N$  se obtiene la siguiente ecuación diferencial ordinaria (EDO) para los coeficientes de la aproximación

$$\begin{aligned} \dot{\boldsymbol{\alpha}} &= \mathbf{P}\boldsymbol{\alpha}, \\ \boldsymbol{\alpha}(T) &= \mathbf{L}^{-1}\mathbf{V}_T, \end{aligned} \quad (3.11)$$

donde:

$$\begin{aligned} \mathbf{P} &= \left( r\mathbf{I} - \frac{1}{2}\sigma^2\mathbf{L}^{-1}\mathbf{L}_{ss} - r\mathbf{L}^{-1}\mathbf{L}_s \right), \\ \boldsymbol{\alpha} &= [\alpha_j(t)], \\ \mathbf{L} &= [\Phi(s_i - s_j)], \\ \mathbf{L}_s &= [s_i\Phi_s(s_i - s_j)], \\ \mathbf{L}_{ss} &= [s_i^2\Phi_{ss}(s_i - s_j)]. \end{aligned} \quad (3.12)$$

Entonces, la aproximación de la solución del PVIBS por medio de funciones de base radial se reduce a la solución de una EDO lineal en  $\mathbb{R}^N$ .

### 3.4 Discretización temporal

Se han aplicado técnicas estándar para resolver la EDO (2.10), por ejemplo, en [12] se realizan pruebas con el método de Euler, Runge-Kutta de orden 2 y 4, un método  $\theta$  implícito, y en [26] se usa un método multipaso implícito.



Denotemos por  $\mathbf{V}^n = \mathbf{L}\boldsymbol{\alpha}^n$  la aproximación de  $V(s_i, T - n\Delta t)$ , para cada paso en el tiempo  $T - n\Delta t$ , entonces,

- Método de Euler

$$\boldsymbol{\alpha}^n = \boldsymbol{\alpha}^{n-1} - \Delta t \mathbf{P} \boldsymbol{\alpha}^{n-1} \quad (3.13)$$

- Método Runge-Kutta explícito de segundo orden

$$\mathbf{F}_1 = -\Delta t \mathbf{P} \boldsymbol{\alpha}^{n-1} \quad (3.14)$$

$$\mathbf{F}_2 = -\Delta t \mathbf{P} (\boldsymbol{\alpha}^{n-1} + 0.5\mathbf{F}_1)$$

$$\boldsymbol{\alpha}^n = \boldsymbol{\alpha}^{n-1} + 0.5(\mathbf{F}_1 + \mathbf{F}_2)$$

- Método Runge-Kutta explícito de cuarto orden

$$\mathbf{F}_1 = -\Delta t \mathbf{P} \boldsymbol{\alpha}^{n-1} \quad (3.15)$$

$$\mathbf{F}_2 = -\Delta t \mathbf{P} (\boldsymbol{\alpha}^{n-1} + 0.5\mathbf{F}_1)$$

$$\mathbf{F}_3 = -\Delta t \mathbf{P} (\boldsymbol{\alpha}^{n-1} + 0.5\mathbf{F}_2)$$

$$\mathbf{F}_4 = -\Delta t \mathbf{P} (\boldsymbol{\alpha}^{n-1} + \mathbf{F}_3)$$

$$\boldsymbol{\alpha}^n = \boldsymbol{\alpha}^{n-1} + \frac{1}{6}(\mathbf{F}_1 + \mathbf{F}_2 + \mathbf{F}_3 + \mathbf{F}_4)$$

- Método  $\theta$  implícito

$$\boldsymbol{\alpha}^n = \boldsymbol{\alpha}^{n-1} - \Delta t \mathbf{P} (\theta \boldsymbol{\alpha}^{n-1} + (1 - \theta) \boldsymbol{\alpha}^n) \quad (3.16)$$

- Método multipaso (BDF2)

En este caso se divide el intervalo de tiempo  $[0, T]$  en  $M$  pasos de tamaño

$\Delta t_n = t_n - t_{n-1}$ ,  $n = 1, \dots, M$ , entonces

$$\begin{aligned}\alpha^1 &= \alpha^0 - \Delta t_1 \mathbf{P} \alpha^1 \\ \alpha^n &= \beta_1^n \alpha^{n-1} - \beta_2^n \alpha^{n-2} - \beta_0^n \mathbf{P} \alpha^n, \quad n = 2, \dots, M\end{aligned}$$

donde

$$w_n = \frac{\Delta t_n}{\Delta t_{n-1}}, \quad \beta_0^n = \Delta t_n \frac{1+w_n}{1+2w_n}, \quad \beta_1^n = \frac{(1+w_n)^2}{1+2w_n}, \quad \beta_2^n = \frac{1+w_n}{w_n^2}.$$

El tamaño de paso variable no es realmente necesario en el PVIBS, en [26] se sugiere imponer condiciones sobre los coeficientes del método,  $\beta_0^n = \beta_0^{n-1}$

$$\Delta t_n \frac{1+w_n}{1+2w_n} = \Delta t_{n-1} \frac{1+w_{n-1}}{1+2w_{n-1}} \iff \frac{1+w_n}{1+2w_n} = \frac{1+w_{n-1}}{1+2w_{n-1}} \equiv c_{n-1},$$

simplificando y resolviendo para la raíz positiva, se obtiene

$$\begin{aligned}w_n &= c_{n-1} - \frac{1}{2} + \frac{\sqrt{4c_{n-1}^2 + 1}}{2}, \quad n = 2, \dots, M, \\ c_n &= \frac{1+w_n}{1+2w_n}, \quad n = 2, \dots, M. \\ c_1 &= 1\end{aligned}$$

Dada la longitud del intervalo de tiempo  $T$ , se elige  $\Delta t_1$ , de tal manera que

$$\sum_{n=1}^M \Delta t_n = \Delta t_1 \left( 1 + \sum_{i=2}^M \prod_{j=2}^i w_n \right) = T. \quad (3.17)$$

Denotemos  $\beta \equiv \beta_0^n$ , entonces el método BDF2 modificado es

$$\begin{aligned}\alpha^1 &= \alpha^0 - \Delta t_1 \mathbf{P} \alpha^1 \\ \alpha^n &= \beta_1^n \alpha^{n-1} - \beta_2^n \alpha^{n-2} - \beta \mathbf{P} \alpha^n, \quad n = 2, \dots, M\end{aligned} \quad (3.18)$$

### Implementación de condiciones de frontera

La implementación de las condiciones de frontera artificiales (3.8), (3.9) es bastante directa. Supongamos que la aproximación de  $V(0, T-n\Delta t)$ ,  $V(\tilde{S}, T-n\Delta t)$  se encuentran en la primera y última posición de  $\mathbf{V}^n$ , respectivamente. Para determinado paso en el tiempo los coeficientes se actualizan como sigue:

1. Calcular

$$\mathbf{V}^n = \mathbf{L}\boldsymbol{\alpha}^n. \quad (3.19)$$

2. Actualizar las condiciones de frontera

$$\begin{aligned} \mathbf{V}^n(1) &= V(0, t) \\ \mathbf{V}^n(N) &= \lim_{S \rightarrow \infty} V(S, t). \end{aligned} \quad (3.20)$$

3. Actualizar los coeficientes

$$\boldsymbol{\alpha}^n = \mathbf{L}^{-1}\mathbf{V}^n. \quad (3.21)$$

## 3.5 Costo computacional

Es importante tomar en cuenta el costo computacional de los algoritmos descritos anteriormente, en la tabla 3.5 se resume el costo total utilizando (3.13) para marcar en el tiempo

Operaciones	Costo	Cantidad
$\mathbf{L}^{-1}$	$O(N^3)$	1
$\mathbf{L}^{-1} \left( r\mathbf{L} - \frac{1}{2}\sigma^2\mathbf{L}_{ss} - r\mathbf{L}_s \right)$	$O(N^3)$	1
$\mathbf{P}\boldsymbol{\alpha}^{n-1}$	$O(N^2)$	$M$
Total	$O(N^3 + MN^2)$	

dónde  $M$  es el número de pasos en el tiempo y  $N$  es el número de puntos espaciales.

Aunque las operaciones más costosas se realizan una vez, conforme  $N$  crece, evaluar la solución numérica del PVIBS de esta manera es una tarea costosa computacionalmente  $O(N^3)$  conforme  $N \rightarrow \infty$ .

# Capítulo 4

## Aceleración de los cálculos

Una mejor estrategia para resolver el PVI (3.11) sería dejar la matriz  $\mathbf{L}$  en (3.13) del lado izquierdo, y en cada paso del tiempo resolver el sistema de ecuaciones lineales correspondiente mediante algún método apropiado.

Por ejemplo, si se utiliza un método iterativo para resolver sistemas de ecuaciones lineales, entonces el costo computacional correspondiente es:

Operaciones	Costo	Cantidad
resolver $\mathbf{L}\alpha^n = \tilde{\alpha}$	$O(kN^2)$	$M$
$\mathbf{P}\alpha^{n-1}$	$O(N^2)$	$M$
Total	$O(kMN^2)$	

donde  $k$  es el número de iteraciones necesarias para resolver el sistema  $\mathbf{L}\alpha^n = \tilde{\alpha}$ . Proceder de esta manera resulta en una mejora sustancial con respecto a  $O(N^3 + MN^2)$ , conforme  $N \rightarrow \infty$ .

Además, se puede acoplar un método rápido para evaluación de FBR, para acelerar las multiplicaciones matriz vector requeridas, tanto en el método iterativo, como en la evaluación del lado derecho para cada paso en el tiempo.

El contenido de este capítulo se distribuye en dos partes principales, en la sección 4.1 se explica el método iterativo utilizado en este trabajo para resolver sistemas de ecuaciones lineales y en la sección 4.2 se resumen el uso del método multipolos y la Transformada Rápida de Gauss en la multiplicación rápida de matrices.

**Nota 4.0.1** *En [26] se utiliza un enfoque ligeramente distinto en donde se explotan simetrías geométricas, tanto para la evaluación de los productos matriz-vector, como para la solución de las ecuaciones.*

## 4.1 Métodos Iterativos

Los sistemas que surgen en problemas de interpolación por medio de FBR, generalmente son mal condicionados. Se conocen dos parámetros que contribuyen de manera importante al mal condicionamiento de estos sistemas, el parámetro de forma, y la distribución espacial de los datos.

En [5] se muestra como el número de condición de las matrices que surgen en la interpolación por medio de FBR multicuádricas crece exponencialmente conforme aumenta el parámetro de forma. También existen varios trabajos donde se estudia cómo varía el número de condición conforme disminuye la mínima distancia entre los puntos  $x_j$ .

En varios trabajos [1, 31] se trata el mal condicionamiento debido a la distribución de los puntos, realizando una interpolación local. Estos métodos consisten de dos etapas, primero, calcular un preconditionador usando funciones cardinales aproximadas, y segundo, el paso iterativo, que generalmente converge en pocas iteraciones. La parte iterativa en estos métodos generalmente tiene

buen desempeño, sobre todo si se acopla un algoritmo rápido para la evaluación de productos matriz vector, pero la tarea de calcular el preconditionador de manera eficiente suele ser una tarea difícil. En [22] se desarrolla un algoritmo rápido ( $O(N \log N)$ ) para la evaluación de este paso.

### 4.1.1 Precondicionamiento con funciones cardinales aproximadas

Las función cardinal  $z^l$  correspondiente a un punto  $x_l$  es una función de la forma

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

que satisface las condiciones de Kronecker

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

es decir,  $z^l$  vale 1 en el  $l$ -ésimo dato, y cero en los demás. Si se conocieran los coeficientes  $\zeta_j^l, j = 1, \dots, N$  para todas las funciones  $z^l$ , calcular la solución del problema de interpolación es bastante simple. Calcular los coeficientes  $\zeta_j^i$  para la función  $z^l$  por un método directo requiere  $O(N^3)$  operaciones, y hay  $N$  de estas funciones.

El enfoque empleado por Beatson y Powell [7] para desarrollar un preconditionador fué usar aproximaciones  $\hat{z}^l$  de estas funciones cardinales, tal que los coeficientes de  $z^l$  se aproximan con un conjunto pequeño, digamos de  $q$  puntos especialmente escogidos. Entonces la matriz  $\hat{T}$  resultante se usa como preconditionador en un algoritmo de subespacios de Krylov.

Dados los conjuntos donde se aproximan las funciones cardinales, el costo de calcular  $\hat{T}$  es de  $O(Nq^3)$ , para  $q \ll N$  e independiente de  $N$ .

En algunos intentos por determinar estos conjuntos, se usaron puntos cercanos al punto correspondiente al índice de la función cardinal, se observó que esta estrategia sólo funciona para conjuntos de datos de tamaño relativamente pequeño, pero no para conjuntos grandes. El algoritmo presentado en [1] es una variación que funcionó en todos los conjuntos de prueba.

### Elección de puntos para aproximar las funciones cardinales

Para construir el preconditionador, es necesario calcular  $N-1$  conjuntos,  $\mathcal{L}_l, l = 1, \dots, N-1$ . Este problema se puede establecer como sigue:

Dado un conjunto  $\mathbb{X} \subset \mathbb{R}^d$ ,  $\mathbb{X} = x_i, i = 1, \dots, N$  y  $q < N$ , construir  $N-1$  conjuntos  $\mathbb{L}_j \subset \mathbb{X}, j = 1, \dots, N-1$  tales que:

- Para  $j=1, \dots, N-q+1$ ,  $\mathbb{L}_j$  consiste de  $q$  puntos  $x_{i_{j1}}, \dots, x_{i_{jq}} \in \mathbb{X}$ , mientras que para  $j = N-q+2, \dots, N-1$ ,  $\mathbb{L}_j$  consiste de  $N-j+1$  puntos  $x_{i_{j1}}, \dots, x_{i_{j,N-j+1}} \in \mathbb{X}$ . Al número de puntos en el conjunto  $\mathbb{L}_j$  se le llama *potencia* (Pow),  $\text{Pow}(\mathbb{L}_j) = q$  para  $j = 1, \dots, N-q+1$  y  $\text{Pow}(\mathbb{L}_j) = N-j+1$  para  $j = N-q+2, \dots, N-1$ .
- Cada conjunto  $\mathbb{L}_j$  se asocia con un punto  $x_{i_{j1}} \in \mathbb{L}_j$ , llamado el centro del conjunto. Entonces se denota a  $\mathbb{C}_j = \{x_{i_{11}}, \dots, x_{i_{j1}}\}, j = 1, \dots, N-1$  como el conjunto de centros.
- El conjunto  $\mathbb{L}_1$  consiste de los puntos  $x_{i_{j1}}, \dots, x_{i_{jq}} \in \mathbb{X}$ , tales que

$$|x_{i_{11}} - x_{i_{12}}| = \min_{x_i, x_k \in \mathbb{X}, i \neq k} |x_i - x_k| \quad (4.1)$$

$$|x_{i_{11}} - x_{i_{1l}}| \leq \min_{x_k \in \mathbb{X} \setminus \mathbb{L}_1} |x_{i_{11}} - x_k|, \text{ para } l = 3, \dots, q. \quad (4.2)$$

Es decir, la distancia entre el centro del conjunto  $\mathbb{L}_1$  y el segundo elemento



es mínima entre todas las distancias entre puntos de  $\mathbb{X}$ , y aparte del centro, el conjunto  $\mathbb{L}_1$  consiste de los  $q - 1$  puntos más cercanos al centro.

- El conjunto  $\mathbb{L}_j, j = 2, \dots, N - 1$  consiste de los puntos  $x_{i_{j1}}, \dots, x_{i_{jp}} \in \mathbb{X} \setminus \mathbb{C}_{j-1}, p = \text{Pow}(\mathbb{L}_j)$ , tales que

$$|x_{i_{j1}} - x_{i_{j2}}| = \min_{x_i, x_k \in \mathbb{X} \setminus \mathbb{C}_{j-1}, i \neq k} |x_i - x_k| \quad (4.3)$$

$$|x_{i_{j1}} - x_{i_{jl}}| \leq \min_{x_k \in \mathbb{X} \setminus \mathbb{L}_j \cup \mathbb{C}_{j-1}} |x_{i_{j1}} - x_k|, \text{ para } l = 3, \dots, p. \quad (4.4)$$

Es decir, la distancia entre el centro del conjunto  $\mathbb{L}_j$  y el segundo elemento es mínima entre todas las distancias entre puntos de  $\mathbb{X}$  excluyendo los centros de los conjuntos  $\mathbb{L}_1, \dots, \mathbb{L}_{j-1}$ , además del centro, el conjunto consiste de los  $p - 1$  puntos más cercanos al centro.

Este problema se puede resolver de manera muy simple calculando las distancias entre todos los puntos, pero a un costo computacional de  $O(N^2)$ .

**Nota 4.1.1** En [22] se explica de manera detallada como construir estos conjuntos de manera eficiente en  $O(N \log N)$  operaciones.

### 4.1.2 El algoritmo iterativo

El método consiste en actualizar la aproximación  $s^k$  de  $f$  en cada iteración,

$$s^k(x) = \sum_{j=1}^N \lambda_j^k \Phi(x - x_j) + \alpha^k, x \in \mathbb{R}^d, \quad (4.5)$$

dados los coeficientes  $\lambda_j^k, j = 1, \dots, N$  y la constante  $\alpha^k$ .

El residual en el paso  $k$  se calcula como

$$r_i^k = f(x_i) - s^k(x_i), i = 1, \dots, N,$$

en los puntos  $x_i$ .

El subespacio de Krylov considerado para este algoritmo es el inducido por el operador  $\Xi$  que actúa sobre las funciones que se pueden expresar en términos de FBR. Este operador se define usando las  $N - 1$  aproximaciones  $\tilde{z}^l$  de las funciones cardinales  $z^l$

$$(\Xi s)(x) = \sum_{l=1}^{N-1} \frac{\langle \tilde{z}_l, s \rangle}{\|\tilde{z}_l\|_\phi^2} \tilde{z}_l(x)$$

donde, el producto interior  $\langle \cdot, \cdot \rangle_\phi$  y la seminorma  $\|\cdot\|_\phi$  se definen como sigue:

Sean  $\lambda, \mu$  los coeficientes resultantes de interpolar las funciones  $s, t$  en los puntos  $x_j$  usando FBR, entonces

$$\begin{aligned} \|s\|_\phi &= -(\lambda^T \Phi \lambda)^{\frac{1}{2}} \\ \langle s, t \rangle_\phi &= \frac{1}{2} (\|s + t\|_\phi^2 - (\|s\|_\phi^2 + \|t\|_\phi^2)) - \frac{1}{2} (\lambda^T \Phi \mu + \mu^T \Phi \lambda) \\ &= -\sum_{j=1}^N \sum_{i=1}^N \lambda_i \phi(x_i - x_j) \mu_j \\ &= -\sum_{i=1}^N \lambda_i t(x_i) = -\sum_{j=1}^N \mu_j s(x_j). \end{aligned} \tag{4.6}$$

**Nota 4.1.2** Si los valores de las funciones  $s, t$  son conocidos en los puntos de interpolación, entonces el producto  $\langle s, t \rangle_\phi$  puede ser evaluado en  $O(N)$  operaciones vía (4.6).

El paso iterativo se resume en forma de algoritmo como sigue:

**Algoritmo (FGP05):**

Definir la aproximación inicial  $s^1$  como

$$\lambda_j^1 = 0, j = 1, \dots, N; \quad \alpha^1 = \frac{1}{2} (\min(f) + \max(f)),$$

de donde obtenemos que el residual inicial es

$$r_i^k = f_i - s^k(x_i), \quad r = \{r_i\}. \tag{4.7}$$

Se utilizan dos funciones expresadas como sumas de FBR:

$$t^k = \sum_{j=1}^N \tau_j^k \Phi(x - x_j), \quad \sum_{j=1}^N \tau_j^k = 0,$$

$$d^k = \sum_{j=1}^N \delta_j^k \Phi(x - x_j), \quad \sum_{j=1}^N \delta_j^k = 0.$$

La función  $t$  se construye usando las funciones cardinales aproximadas en los puntos escogidos por el algoritmo descrito en la sección 4.1.1 y el valor del residual en la iteración  $k$ :

$$\mu_l^k = \frac{1}{\zeta_l} \sum_{i \in \mathcal{L}_l} \zeta_{li} r_i^k, \quad \tau_j^k = \mu_l^k \zeta_{lj}.$$

Sea  $s^*$  la función que resulta al final del proceso iterativo, en [1] se prueba que

$$t^k = \Xi(s^* - s^k). \quad (4.8)$$

y que la función  $d^k$  satisface

$$d^1 = t^1, \quad d^k = t^k - \frac{\langle t^k, d^{k-1} \rangle_\phi}{\langle d^{k-1}, d^{k-1} \rangle_\phi} d^{k-1}, \quad (4.9)$$

de la relación (4.9) se puede deducir que se satisface el criterio de conjugación  $\langle d^k, d^{k-1} \rangle_\phi = 0$ .

Una vez calculados los coeficientes  $\tau_j, \delta_j$ , la solución se actualiza con un tamaño de paso  $\gamma_k$  a lo largo de la dirección  $d^k$  como sigue

$$\lambda_j^{k+1} = \lambda_j^k + \gamma_k d^k, \quad \gamma_k = \frac{\sum_{i=1}^N \delta_i^k r_i^k}{\sum_{i=1}^N \delta_i^k d^k(x_i)} \quad (4.10)$$

y la constante  $\alpha^k$  se elige de tal manera que minimice el elemento máximo en el nuevo vector residual. Se puede probar que la cantidad  $\gamma^k$  es tal que minimiza el residual  $\|s^* - s^{k+1}\|$ .

## 4.2 Multiplicación Rápida de Matrices

Existen varios métodos para evaluar de manera eficiente las sumas de FBR

$$s(y) = \sum_{j=1}^N \lambda_j \Phi(y - x_j), x \in \mathbb{R}^d, \quad (4.11)$$

requeridas, por ejemplo, en la solución del problema de interpolación por medio del algoritmo FGP05 descrito en la sección 4.1.2, entre ellos se encuentran el Método Multipolos y un método basado en la Transformada Rápida de Gauss (TRG).

El método Multipolos tuvo origen en el trabajo de L. Greengard y V. Rokhlin [20] (1987) sobre la simulación rápida de sistemas de partículas, en el cuál se desarrolló un método para sumar de manera eficiente una FBR que surge en estudio de campos de fuerzas gravitacionales ó Coulombicas. Este método ha sido adaptado de manera exitosa a otras FBR [32, 6]. El método Multipolos logra reducir el costo de evaluar un producto matriz vector de  $O(N^2)$  a  $O(N \log N)$  operaciones.

En 2005, G. Roussos y B. Baxter [18] desarrollaron un método para la suma rápida de FBR multicuádricas y de placa delgada, basado en la Transformada Rápida de Gauss [21], que logra la tarea de realizar el producto matriz vector en un costo de  $O(N)$  operaciones, para las FBR mencionadas.

Existen varias limitaciones y ventajas de cada uno de estos métodos:

- El método Multipolos se puede aplicar a una amplia variedad de FBR, pero generalmente la eficiencia se deteriora de manera importante en dimensiones altas, y la implementación eficiente de este método requiere una cantidad de trabajo bastante considerable.

- El método de Roussos aplica a un conjunto más restringido de FBR, pero tiene costo computacional conforme  $N \rightarrow \infty$ .

### 4.2.1 Suma rápida de Multicuádras con el Método Multipolos

El método multipolos consta de tres componentes principales.

- Expansiones de campo lejano rápidamente convergentes para la función  $\Phi(x - y)$ .
- División jerárquica del espacio en celdas o paneles de puntos.
- Fórmulas para trasladar de manera eficiente el centro de las expansiones.

#### Expansiones

En [6] se desarrollan expansiones de campo lejano para la evaluación de FBR multicuádras,

$$s(y) = \sum_{i=1}^N \lambda_i (\|y - x_i\|^2 + c^2)^{k/2}, \quad y \in \mathbb{R}^d, \quad (4.12)$$

para  $k$  impar, y  $c \geq 0$ .

**Teorema 4.2.1** Sean  $x_i \in \mathbb{R}^d$ ,  $\|x_i\| \leq r$  y  $\lambda_i \in \mathbb{R}$ . Sea  $k$  impar,  $c \geq 0$ ,  $s$  dada por (4.12) y

$$\begin{aligned} Q_l(y) &= \sum_{i=1}^N \lambda_i P_l^{(k)}(\|x_i\|^2 + c^2, -2\langle x_i, y \rangle, \|y\|^2), \quad l \in \mathbb{N}_0, \\ s_p(y) &= \sum_{l=0}^{p+k} \frac{Q_l(y)}{\|y\|^{2l-k}}, \quad y \in \mathbb{R}^d \setminus \{0\}. \end{aligned} \quad (4.13)$$

Entonces para  $y$  tal que  $\|y\| > R = \sqrt{r^2 + c^2}$ ,

$$\begin{aligned} |s(y) - s_p(y)| &\leq 2^k MR^k \left(\frac{1}{\tau}\right)^{p+1} \frac{1}{1 - 1/\tau}, \quad k > 0 \\ &\leq \binom{p}{p+k+1} MR^k \left(\frac{1}{\tau}\right)^{p+1} \left(\frac{1}{1 - 1/\tau}\right)^{-k}, \quad k < 0, \end{aligned}$$

donde  $M = \sum_{i=1}^N |\lambda_i|$  y  $\tau = \|y\|/R$ , y  $P_l^{(k)}$  está dado por

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

Además, se cuenta con una relación de recurrencia para evaluar de manera eficiente los polinomios  $P_l^{(k)}$

$$(l+1)P_{l+1}^{(k)}(u, v, w) = \left(\frac{k}{2} - l\right) v P_l^{(k)}(u, v, w) + (k - (l-1)) uw P_{l-1}^{(k)}(u, v, w).$$

### Traslación de expansiones

Es posible trasladar el centro de una expansión de campo lejano conociendo sólo sus coeficientes, sin conocimiento directo de los correspondientes puntos  $x_i$  y pesos  $\lambda_i$ .

Sea

$$s_p(y) = \sum_{l=0}^{p+k} \frac{Q_l(x)}{\|x\|^{2l-k}}, \quad x = y - t \neq 0, \quad (4.14)$$

una expansión similar a (4.13), pero centrada en  $t \neq 0$ . Se desea trasladar el centro de esta expansión, esto es, buscamos polinomios homogéneos  $\hat{Q}_l$ , de grado  $l$ , tales que

$$s_p(y) = \sum_{l=0}^{p+k} \frac{\hat{Q}_l(y)}{\|y\|^{2l-k}} + O(\|y\|^{-(p+1)}), \quad (4.15)$$

cuando  $\|y\| \rightarrow \infty$ . La translación de expresiones como (4.14) en expansiones de la forma (4.15) se puede realizar en  $O((p+k)^{n+1})$  operaciones, y se puede

implementar de manera eficiente en  $O((p+k)^n \log(p+k))$  vía convolución por medio de transformada rápida de Fourier.

Sea

$$Q(x) = \sum_{l=0}^{p+k} Q_l(x) \|x\|^{2(p+k-l)},$$

entonces,

$$s_p(y) = \frac{Q(x)}{\|x\|^{2p+k}}, \quad x = y - t \neq 0. \quad (4.16)$$

Dado que se conocen los coeficientes  $Q_l$ , para obtener  $Q$  sólo hace falta formar los polinomios  $\|\cdot\|^{2(p+k-l)}$  y luego formar los productos  $Q_l(\cdot)\|\cdot\|^{2(p+k-l)}$ , cuyo grado no es mayor que  $2(p+k)$ . Se puede calcular cada uno de estos productos en  $O((p+k)^{n-1} \log(p+k))$  operaciones. Dado que hay  $p+k+1$  de estos productos, para formar  $Q$ , se requieren  $O((p+k)^n \log(p+k))$  operaciones.

Sea

$$\tilde{Q}(y) = Q(y-t), \quad (4.17)$$

la siguiente tarea es reescribir  $\tilde{Q}$  en una suma de productos de potencias de  $y$  y polinomios homogéneos. En [6] se argumenta que  $\tilde{Q}$  se puede reescribir como

$$\tilde{Q}(y) = \sum_{l=0}^{p+k} q_l(y) \|y\|^{2(p+k-l)} + q_{\text{low}}(y) \quad (4.18)$$

donde  $q_l$  son polinomios homogéneos de grado  $l$  y  $q_{\text{low}}$  es algún polinomio de grado  $p+k-1$  o menor.

De (4.16) y (4.17), se tiene

$$\begin{aligned}
s_p(y) &= Q(y-t)\|y-t\|^{-(2p+k)} = \tilde{Q}(y)\|y-t\|^{-(2p+k)} \\
&= \tilde{Q}(y) \sum_{m=0}^{\infty} P_m^{(-2p-k)}(\|t\|^2, -2\langle y, t \rangle, \|y\|^2) \frac{1}{\|t\|^{2p+k+2m}} \\
&= \left( \sum_{l=0}^{p+k} q_l(y)\|y\|^{2(p+k-l)} + q_{\text{low}}(y) \right) \\
&\quad \times \left( \sum_{m=0}^{\infty} P_m^{(-2p-k)}(\|t\|^2, -2\langle y, t \rangle, \|y\|^2) \frac{1}{\|y\|^{2p+k+2m}} \right) \\
&= \sum_{l=0}^{p+k} \sum_{m=0}^{\infty} q_l(y) P_m^{(-2p-k)}(\|t\|^2, -2\langle y, t \rangle, \|y\|^2) \frac{1}{\|y\|^{2(m+l)-k}} + O(\|y\|^{2(-p-1)}) \\
&= \sum_{l=0}^{p+k} \left( \sum_{j=0}^l q_j(y) P_{l-j}^{(-2p-k)}(\|t\|^2, -2\langle y, t \rangle, \|y\|^2) \right) \frac{1}{\|y\|^{2l-k}} + O(\|y\|^{2(-p-1)}) \\
&= \sum_{l=0}^{p+k} \hat{Q}_l(y) \frac{1}{\|y\|^{2l-k}} + O(\|y\|^{2(-p-1)}).
\end{aligned}$$

dónde  $\hat{Q}_l(y)$  se puede calcular por medio de una convolución de tamaño  $p+k+1$  vía transformada rápida de Fourier en  $O((p+k)^n \log(p+k))$  operaciones.

### Caso unidimensional

Nótese que para  $d=1$ , (4.13) se reduce a

$$\begin{aligned}
s_p(y) &= \text{sign}(y) \sum_{l=0}^{p+k} \frac{Q_l}{y^{l-k}}, \tag{4.19} \\
Q_l &= \sum_{i=1}^N \lambda_i P_l^{(k)}(x_i^2 + c^2, -2x_i, 1).
\end{aligned}$$

Denotemos  $r_T, s_T$  la aproximación de campo lejano dada por (4.13) y la spline debidas a  $x_i \in T = [-h, h)$ , respectivamente centradas en  $y=0$ . Para obtener una expansión centrada en  $y=t$  en lugar de  $y=0$ , para el panel



$T = [t - h, t + h)$ , se hace el cambio de variable correspondiente

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

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

La aproximación trasladada converge para  $y$  suficientemente lejos de  $T$  (al menos el diámetro de  $T$ ), en cuyo caso se dice que  $y$  está bien separado de  $T$ .

En [32] se describe cómo se usan estas expansiones de campo lejano en una estructura de datos jerárquica para llevar a cabo la evaluación eficiente de  $s$ . A continuación se describe brevemente para  $d = 1$ .

### División del espacio

Supongamos que los puntos  $x_i$  están contenidos en el intervalo unitario  $[0, 1]$ , el cual se subdivide en un árbol binario de celdas de puntos, como lo ilustra la figura 4.1.

El siguiente paso es calcular expansiones correspondientes a cada una de las celdas en el árbol, excluyendo la raíz.

Para evaluar  $s(y)$  se desciende en el árbol, de manera que se sume la contribución de las celdas  $T$  que están bien separadas de  $y$  mediante la aproximación  $r_T(y)$ , y dejar hasta el final las que no están bien separadas, para sumar su contribución de manera directa mediante  $s_T(y)$ .

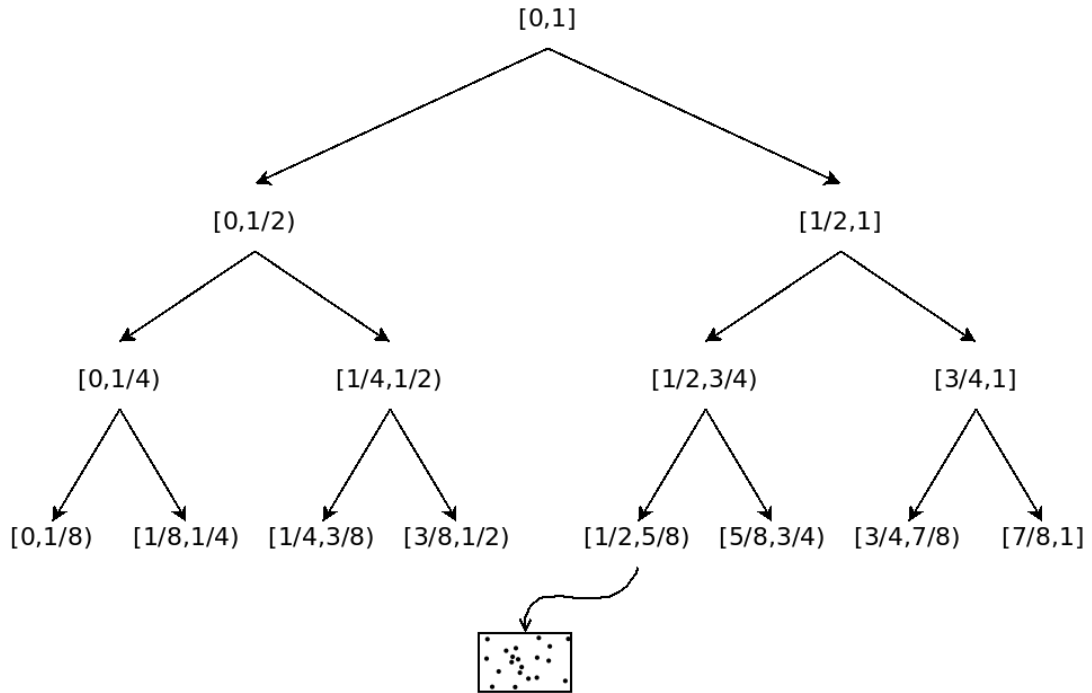


Figura 4.1: Subdivisión jerárquica del espacio.

Por ejemplo, para evaluar en  $y \in [\frac{3}{8}, \frac{1}{2})$ , se utiliza la siguiente expresión

$$\begin{aligned}
 s(y) &\approx r_{[3/4,1)}(y) + r_{[0,1/8)}(y) + r_{[1/8,1/4)}(y) \\
 &+ s_{[1/4,3/8)}(y) + s_{[3/8,1/2)}(y) + s_{[1/2,5/8)}(y) \\
 &+ r_{[5/8,3/4)}(y)
 \end{aligned}$$

### El algoritmo de evaluación

El algoritmo del método Multipolos se puede resumir en dos pasos, primero, la construcción de una estructura de datos, y segundo, el proceso de evaluación:

### 1 - Estructura de datos

Dados los puntos  $\{x_i\}_{i=1}^N$ , los coeficientes  $\{\lambda_i\}_{i=1}^N$  y la precisión requerida  $0 < \epsilon < \frac{1}{2}$ .

- Elegir  $p \approx |\log_{2.12}(\epsilon/\|\boldsymbol{\lambda}\|_1)|$ , para asegurar que el error está dentro de la precisión requerida.
- Elegir el número de niveles  $m \approx \log_2 N$ , esto asegura que el número de puntos en cada celda está acotado independientemente de  $N$ .
- Iniciar con  $[0, 1]$ , subdividiendo cada celda en dos partes, hasta llegar al nivel  $m$ , e ir asignando los  $x_i$  correspondientes.
- Ascender desde el nivel  $m$  hasta el nivel 2, calculando las expansiones  $r_T$  correspondientes.

Alternativamente se pueden calcular sólo las expansiones del nivel más bajo, y ascender trasladando los centros de las expansiones del nivel más bajo (esto reduce significativamente el costo de esta parte del algoritmo, ya que la traslación indirecta depende sólo de la longitud de la serie, y no del número de puntos en una celda).

### 2 - Evaluación

- Dado  $y$ , se desciende en el árbol, desde el nivel 2, hasta el nivel  $m$ , sumando la influencia aproximada de las celdas bien separadas de  $y$  (al acumular la influencia de una celda en el nivel más alto posible, se ahorra mayor cantidad de trabajo), y se deja hasta el último nivel las celdas que no están bien separadas de  $y$ , para las cuales se suma de forma directa.

El paso de evaluación se puede programar de manera muy simple usando recursión. En el siguiente pseudocódigo, las celdas se denotan por  $T$ , y sus hijos izquierdo y derecho, respectivamente, se denotan por  $T \rightarrow \text{izq}$ ,  $T \rightarrow \text{der}$ .

```
eval(T,y)
{
  if (T esta separado de y)
    retorna r_T(y)
  else if (T no tiene hijos)
    retorna s_T(y)
  else
    retorna eval(T->izq,y)+eval(T->der,y)
}
```

## 4.2.2 Suma Rápida de Multicuádras usando la Transformada Rápida de Gauss

A continuación se describe el método presentado en [18]. Este método consta de dos partes fundamentales, primero, está basado en cierta representación integral de FBR que involucran a la función exponencial, segundo, que existen métodos rápidos para sumar FBR gaussianas.

Sean  $\{x_i\}_{i=1}^N \subset \mathbb{R}^d$ ,  $y \in \mathbb{R}^d$ , entonces

$$\begin{aligned} s(y) &= \sum_{i=1}^N \frac{\lambda_i}{\sqrt{\|y - x_i\|^2 + c^2}} \\ &= \frac{1}{\sqrt{\pi}} \int_0^\infty \frac{e^{-sc^2}}{\sqrt{t}} \sum_{i=1}^N \lambda_i e^{-s\|y-x_i\|^2} ds, \end{aligned} \quad (4.20)$$

$$\begin{aligned}
s(y) &= \sum_{i=1}^N \lambda_i \sqrt{\|y - x_i\|^2 + c^2} \\
&= c + \frac{1}{2\sqrt{\pi}} \int_0^\infty \frac{e^{-sc^2}}{\sqrt{s}} \left( \frac{\sum_{i=1}^N \lambda_i - \sum_{i=1}^N \lambda_i e^{-s\|y-x_i\|^2}}{s} \right) ds, \quad (4.21)
\end{aligned}$$

Este algoritmo de evaluación rápida se reduce evaluar una discretización de las integrales anteriores, por ejemplo, para la función multicuádrica inversa, si usamos una cuadratura Gauss-Laguerre generalizada [2] de  $q$  términos con  $\alpha = -1/2$ , la expresión discretizada correspondiente es

$$s(y) \approx \frac{1}{\sqrt{\pi}} \sum_{k=1}^q w_k f(s_k), \quad (4.22)$$

$$f(s) = \sum_{i=1}^N \lambda_i e^{-s\|y-x_i\|^2}, \quad (4.23)$$

ahora el trabajo es, sumar  $q$  FBR Gaussianas a un costo total de  $O(qN)$  mediante la TRG, en lugar de una Multicuádrica a un costo  $O(N^2)$ .

**Nota 4.2.1** En [18] se utiliza la TRG de Greengard y Strain [21], aunque el costo de evaluar  $f(s)$  por este medio es  $O(N)$ , deja de ser viable para altas dimensiones debido a que utiliza la metodología Multipolos [11].

De manera más precisa:

- El número de términos en las expansiones de Hermite crece exponencialmente con respecto a la dimensión  $d$ , debido a esto el término constante en el costo  $O(N)$  crece exponencialmente con respecto a la dimensión, además los requerimientos de memoria también resultan afectados de la misma manera.
- El método de subdivisión del espacio consiste en dividir de manera uniforme el cubo, el cual es muy ineficiente en altas dimensiones.

### Transformada Rápida de Gauss Mejorada (TRGM)

En [11] se desarrolla un método para evaluar sumas de Gaussianas eficiente en altas dimensiones ( $d > 3$ ). La idea es utilizar una factorización más sencilla, y explotar el hecho de que no es necesario sumar contribuciones de campo lejano, debido a la forma tan rápida que decae la función exponencial, junto con un método eficiente para la división del dominio.

### Expansiones

Para ser congruentes con la notación de [11], sea

$$G(y_j) = \sum_{i=1}^N q_i e^{-\|y_j - x_i\|^2/h^2}. \quad (4.24)$$

Primero, se reescribe el término exponencial como

$$e^{-\|y_j - x_i\|^2/h^2} = e^{-\|\Delta y_j\|^2/h^2} e^{-\|\Delta x_i\|^2/h^2} e^{2\Delta y_j \cdot \Delta x_i/h^2} \quad (4.25)$$

donde

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

En la expresión (4.25), las primeras dos exponenciales se pueden evaluar de manera individual a un costo de  $O(N)$ . El problema restante es evaluar el tercer término, donde ambos  $y_j$  y  $x_i$  están involucrados. Una manera de romper este acoplamiento es expandir en serie

$$e^{2\Delta y_j \cdot \Delta x_i/h^2} = \sum_{n=0}^{\infty} \Phi_n(\Delta y_j) \Psi_n(\Delta x_i). \quad (4.26)$$

Denotemos  $\phi(\Delta y_j) = e^{-\|\Delta y_j\|^2/h^2}$ ,  $\psi(\Delta x_i) = e^{-\|\Delta x_i\|^2/h^2}$ , entonces la suma (4.24) se puede escribir como

$$G(y_j) = \sum_{i=1}^N q_j \phi(\Delta y_j) \psi(\Delta x_i) \sum_{n=0}^{\infty} \Phi_n(\Delta y_j) \Psi_n(\Delta x_i). \quad (4.27)$$

Si la serie (4.24) converge absolutamente, se puede truncar después de  $p$  términos, entonces intercambiando el orden de las sumas en (4.27), se obtiene

$$G(y_j) = \phi(\Delta y_j) \sum_{n=0}^{p-1} C_n \Phi_n(\Delta y_j) + \epsilon(p) \quad (4.28)$$

$$C_n = \sum_{i=1}^N q_i \psi(\Delta x_i) \Psi_n(\Delta x_i). \quad (4.29)$$

La factorización (4.28) es la base de la TRGM.

Para calcular las funciones y coeficientes en (4.28), (4.29), se usa la siguiente expansión de la función exponencial de una variable

$$e^v = \sum_{n=0}^{\infty} \frac{v^n}{n!}.$$

Para  $v = 2x \cdot y$ , se tiene que

$$e^v = \sum_{|\alpha|=n} 2^n \binom{n}{\alpha} x^\alpha y^\alpha \frac{v^n}{n!},$$

donde  $\alpha$  es un multi-índice y  $\binom{n}{\alpha} = \frac{n!}{\alpha_1! \alpha_2! \dots \alpha_d!}$  son los coeficientes multinomiales, entonces

$$e^{2x \cdot y} = \sum_{\alpha \geq 0} \frac{2^{|\alpha|}}{\alpha!} x^\alpha y^\alpha. \quad (4.30)$$

De las ecuaciones (4.25), (4.28) y (4.30),  $G(y_j)$  se puede expandir en serie de Taylor alrededor de  $x^*$  como sigue

$$G(y_j) = \sum_{\alpha \geq 0} C_\alpha e^{-\|y_j - x^*\|^2/h^2} \left( \frac{y_j - x^*}{h} \right)^\alpha, \quad (4.31)$$

donde los coeficientes están dados por

$$C_\alpha = \frac{2^{|\alpha|}}{\alpha!} \sum_{i=1}^N q_i e^{-\|x_i - x^*\|^2/h^2} \left( \frac{x_i - x^*}{h} \right)^\alpha. \quad (4.32)$$

**Nota 4.2.2** *Si se trunca la serie después del término de grado  $p-1$ , el número de términos  $r_{p-1,d} = \binom{p+d-1}{d}$  es mucho menor que  $p^d$  en altas dimensiones. Por ejemplo, si  $d = 12$  y  $p = 10$ , la versión original de la TRG requiere  $10^{12}$  términos, mientras que en esta nueva estrategia sólo se requieren 293930. Para  $d \rightarrow \infty$  y  $p$  moderado, el número de términos requerido por la TRGM es  $O(d^p)$  lo cual representa una mejor sustancial con respecto a  $p^d$  en la TRG.*

### División del espacio

Para usar la expansión (4.31), se debe dividir el dominio en celdas, de tal forma que al evaluar  $G(y_j)$  se colecte la influencia de los  $x_i$  que pertenecen a sólo unas cuantas celdas vecinas.

Un algoritmo ideal para partir el dominio deberá tener las siguientes características:

- El número de celdas es independiente de la dimensión.
- Que realice la división de manera adaptativa, es decir, de acuerdo a la distribución de los puntos.
- Que cada celda sea lo más compacta posible (una celda ideal es una esfera, pero entonces no se podría dividir el dominio de manera disjunta para puntos arbitrarios)

Basado en estas consideraciones se modela el problema de la división del dominio como el problema de los  $k$ -centros: dado un conjunto de  $N$  puntos y un número predefinido de celdas  $K$ , hallar una partición  $S_1, S_2, \dots, S_K$ , y centros  $c_1, c_2, \dots, c_K$  que minimicen el radio máximo de las celdas:



$$\max_i \max_{v \in S_i} \|v - c_i\|. \quad (4.33)$$

Gonzalez [19] desarrolló un algoritmo bastante sencillo que resuelve este problema de manera aproximada, con un factor de aproximación de 2.

**Algoritmo:**

- Elegir de manera arbitraria un punto  $v_0$  como el centro de la primera celda y agregarlo al conjunto de centros  $C$ .
- Para  $i = 1, \dots, K$ , calcular la distancia de cada punto al conjunto  $C$ ,  $d(v_j, C) = \min_{c \in C} \|v - c\|$ . Entonces agregar a  $C$  el punto más lejano, es decir, el punto  $v_i$  tal que  $d(v_i, C) = \max_{v_j} d(v_j, C)$ . Al final se obtiene  $C = \{v_0, \dots, v_{k-1}\}$ .
- Asignar los puntos restantes a la celda con el centro más cercano.

Una implementación directa tiene un costo de  $O(Nk)$ , pero existen algoritmos eficientes [33] que permiten realizar la división del espacio en  $O(N \log(k))$  operaciones.

**Ejemplo 4.2.1** *En la figura 4.2 se muestra una corrida del algoritmo de división del espacio, para puntos aleatorios uniformemente distribuidos en un cuadrado unitario en el plano ( $d = 2$ ).*

**El algoritmo de evaluación**

A continuación se listan los pasos requeridos en este método rápido para la evaluación de sumas de FBR Gaussianas.

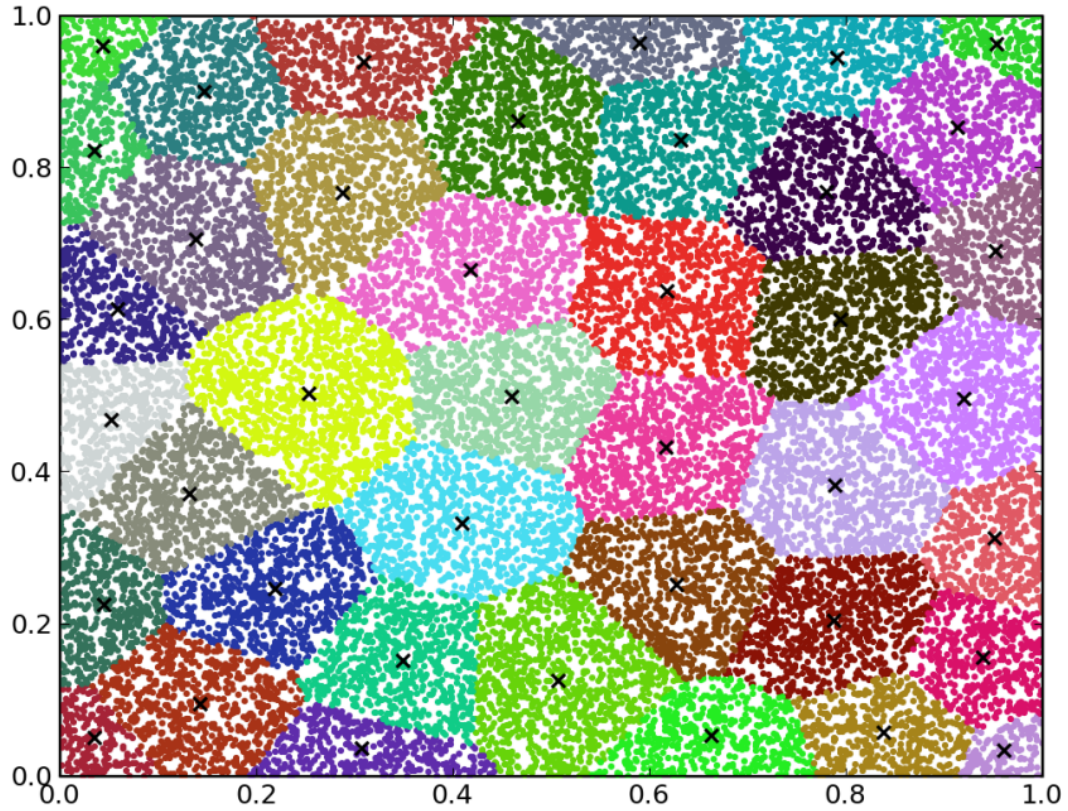


Figura 4.2: División con  $N = 25000$ ,  $K = 40$ , tiempo requerido  $t = 0.026s$ .

**Partición:** Asignar los  $N$  puntos  $x_i, i = 1, \dots, N$  en  $K$  celdas, tal que el radio máximo es menor que  $h\rho_x$ .

**Número de truncamiento:** Elegir  $p$  suficientemente grande tal que el error estimado es menor que la precisión requerida  $\epsilon$ .

**Coefficientes:** Para cada celda  $S_k$ , con centro  $c_k$ , calcular los coeficientes de la expresión (4.32):

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

**Evaluación** Para cada  $y_j$ , hallar las celdas vecinas cuyos centros están en el

rango  $h\rho_y$ . Entonces evaluar (4.24) por medio de la expansión (4.31) como sigue:

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

El costo computacional de los pasos anteriores es:

Paso	Costo
1	$O(NK)$
2	–
3	$O(Nr_{pd})$
4	$Mnr_{pd}$
Total	$O(N(K + r_{pd}) + Mnr_{pd})$

donde  $n$  es el número máximo de vecinos cercanos usados en la evaluación de  $G(y_j)$ .

### Estimación del error

El error al evaluar la TRGM proviene de dos fuentes, debido al truncamiento de la serie (4.31) en el paso 3, y de ignorar los coeficientes de clusters lejanos en el paso 4.

Sea  $E_T(x, p)$  el error debido al truncamiento de la serie después del orden  $p - 1$ , se satisface la cota:

$$\begin{aligned} |E_T(x, p)| &\leq e^{-(\|\Delta x\|^2 + \|\Delta y\|^2)/h^2} \frac{1}{p!} e^{2\Delta x \cdot \Delta y/h^2} \left( \frac{2\Delta x \cdot \Delta y}{h^2} \right)^p \\ &\leq \frac{2^p}{p!} \rho_x^p \rho_y^p \end{aligned}$$

donde  $\rho_x = r_x/h$ ,  $\rho_y = r_y/h$  y  $r_x, r_y$  son cotas superiores para  $\Delta_x, \Delta_y$ , respectivamente.

Ahora, sea  $E_C(r_y)$  el error debido a ignorar las contribuciones fuera del radio  $r_y$ , entonces se satisface:

$$|E_C(r_y)| \leq e^{-\rho_y^2},$$

así el error total al evaluar en el punto  $y$  satisface:

$$|E(y)| \leq Q \left( \frac{2^p}{p!} \rho_x^p \rho_y^p + e^{-\rho_y^2} \right).$$

donde  $Q = \sum |q_j|$ .

Los parámetros  $\rho_x$ ,  $\rho_y$  y  $p$  controlan la precisión del algoritmo de evaluación.

**Ejemplo 4.2.2** *A continuación se muestra el tiempo de ejecución de la TRGM en distintos escenarios, para puntos aleatorios uniformemente distribuidos en el hipercubo unitario y fijando la precisión ( $\approx 10^{-7}$ ).*

**Nota 4.2.3** *En las gráficas de las figuras 4.3, 4.4 se puede observar que el desempeño de la TRGM disminuye con el parámetro  $h$ , esto se debe a que se divide mucho el dominio, entonces para valores de  $h$  muy pequeños es preferible combinar la TRGM con un método rápido para buscar vecinos cercanos.*

*En la figura 4.5 se muestra el tiempo de ejecución TRGM para valores de  $h$  pequeños, usando la librería ANN [14] para búsqueda de vecinos cercanos.*

**Nota 4.2.4** *La implementación de la TRGM que se utilizó en este trabajo es una versión debida Vlad I. Morariu, originalmente desarrollada en C++ por Vikas C. Raykar y Changjiang Yang, la cual está disponible bajo la licencia LGPL. Los algoritmos se corrieron en una pc de escritorio con un AMD64X2 3600+ y 2GB de memoria.*

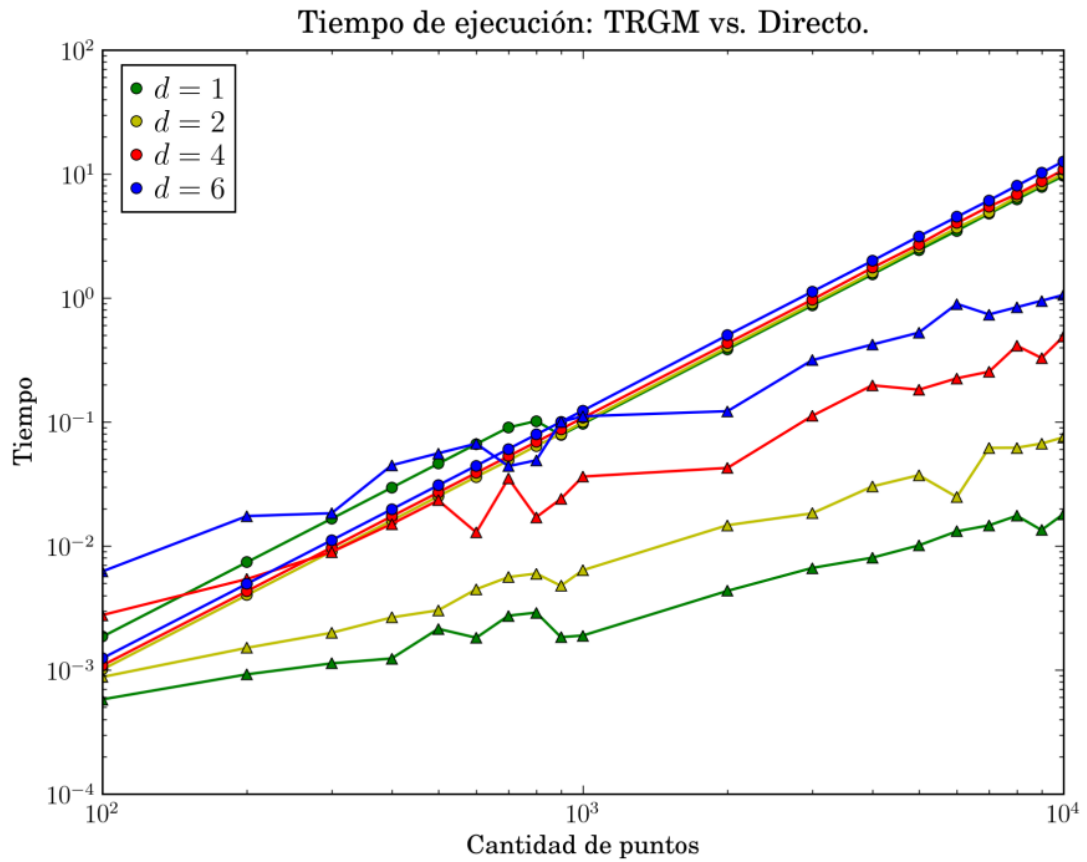


Figura 4.3: Tiempo de ejecución ( $h = 0.5d$ )  $\circ$ -directa,  $\triangle$ -TRGM.

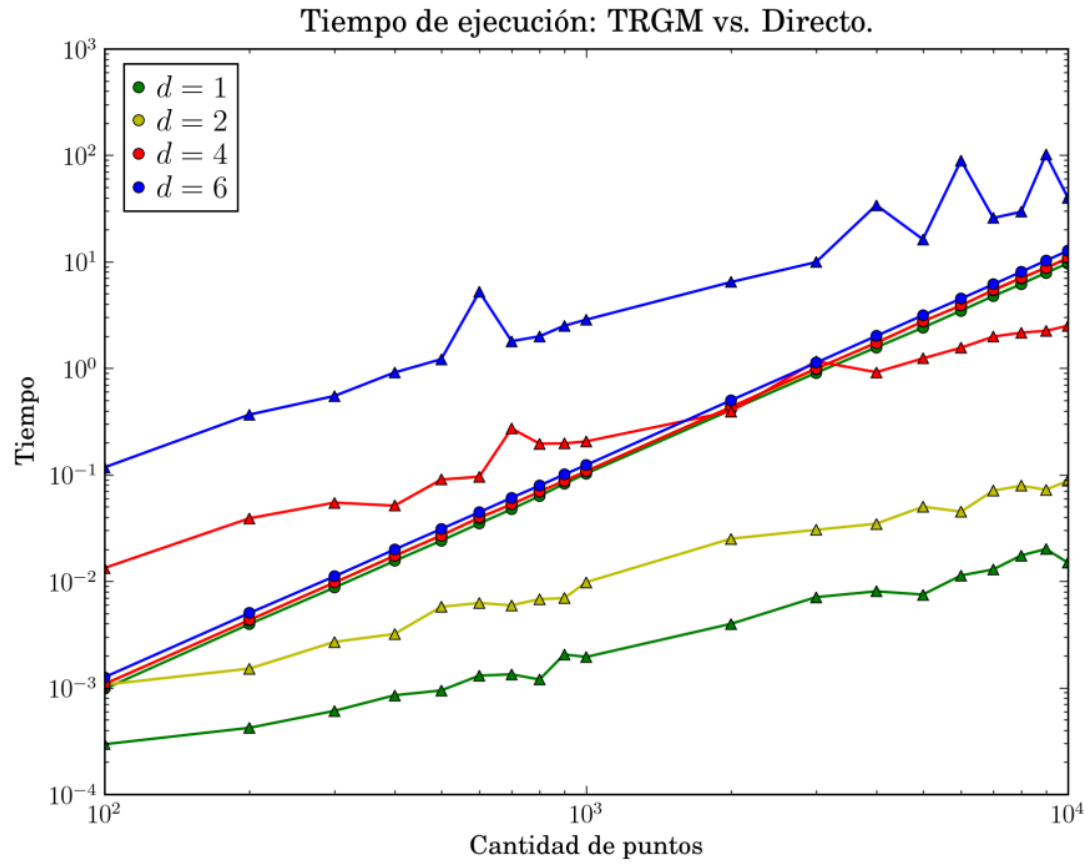


Figura 4.4: Tiempo de ejecución ( $h = 0.5\sqrt{d}$ )  $\circ$ -directa,  $\triangle$ -TRGM.

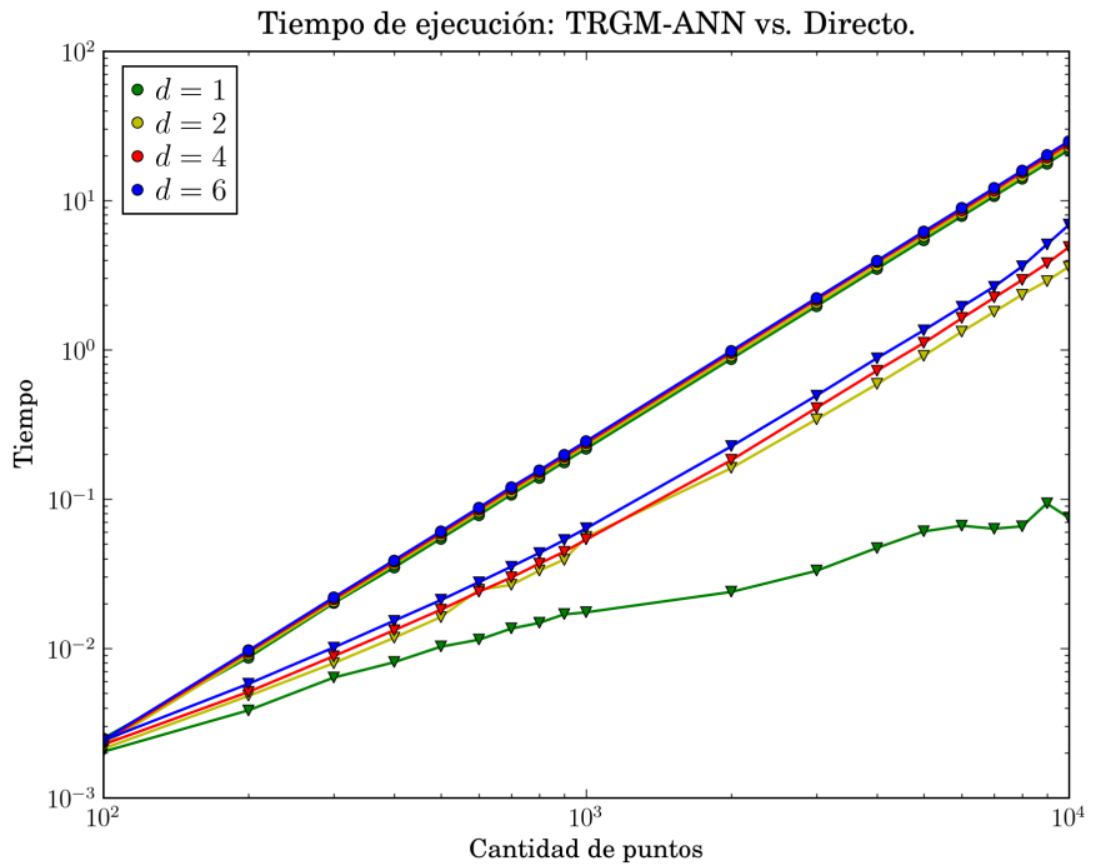


Figura 4.5: Tiempo de ejecución ( $h = 0.001d$ )  $\circ$  directa,  $\triangle$  TRGM-ANN.

# Capítulo 5

## Resultados

Como se mencionó en el capítulo 4, una mejor estrategia para resolver la EDO (3.11) sería dejar la matriz  $\mathbf{L}$  en (3.13) del lado izquierdo, y en cada paso del tiempo resolver el sistema de ecuaciones lineales correspondiente. En este capítulo se realiza una exploración numérica de los algoritmos discutidos en los capítulos anteriores aplicados en la solución de PVIBS.

El contenido de este capítulo está organizado como sigue: en la sección 5.1 muestra cómo se puede adaptar el algoritmo iterativo FGP05 descrito en la sección 4.1 para resolver el PVIBS en combinación con el método implícito BDF2 para marchar en el tiempo. En la sección 5.2 se muestran algunos detalles y resultados numéricos sobre la implementación del método multipolos utilizada en este trabajo. En la sección 5.3 se muestran comparaciones del tiempo de ejecución para la solución del PVIBS empleando el método multipolos para la multiplicación de matrices. Por último, en la sección 5.4 se mencionan detalles sobre la suma rápida de FBR multcuádricas basada en la TRGM.



**Nota 5.0.5** *Se realizaron varias pruebas para los métodos de marcha en el tiempo de la sección 3.4 en la solución directa del PVIBS, se observó que los métodos explícitos (3.13), (3.14) y (3.15) requieren muchos pasos de tiempo conforme aumenta el número de puntos espaciales, mientras que los métodos implícitos (3.16), (3.18) mostraron mejor desempeño. Debido al mal condicionamiento de la matriz  $\mathbf{L}$ , para los ejemplos realizados en este trabajo se usa el método BDF2.*

## 5.1 Solución del PVIBS usando métodos iterativos

**Nota 5.1.1** *En vista de poder utilizar el algoritmo FGP05 para la solución del PVIBS, se debe agregar una constante al interpolante (3.10), dado que la FBR multicuádrica es condicionalmente positivo definida de orden 1.*

El sistema de ecuaciones diferenciales ordinarias que resulta después de dejar la matriz  $\mathbf{L}$  del lado izquierdo en (3.11) es

$$\begin{aligned}\mathbf{L}\dot{\boldsymbol{\alpha}} + \dot{\boldsymbol{\lambda}} &= \mathbf{P}\boldsymbol{\alpha} + r\boldsymbol{\lambda}, \\ \mathbf{L}\boldsymbol{\alpha}(T) + \boldsymbol{\lambda}(T) &= \mathbf{V}_T,\end{aligned}\tag{5.1}$$

precisión donde:

$$\mathbf{P} = r\mathbf{L} - r\mathbf{L}_s - \frac{1}{2}\sigma^2\mathbf{L}_{ss}\tag{5.2}$$

$$\begin{aligned}\mathbf{L} &= \left[ \sqrt{(s_i - s_j)^2 + c^2} \right] \\ \mathbf{L}_s &= \left[ \frac{s_i(s_i - s_j)}{\sqrt{(s_i - s_j)^2 + c^2}} \right] \\ \mathbf{L}_{ss} &= \left[ \frac{s_i^2 c^2}{\left( \sqrt{(s_i - s_j)^2 + c^2} \right)^3} \right]\end{aligned}$$

Al aplicar el algoritmo BDF2 a (5.1), se obtiene el siguiente esquema para marchar en el tiempo

$$\mathbf{L}(\boldsymbol{\alpha}^1 - \boldsymbol{\alpha}^0) + \lambda^1 - \lambda^0 = -\Delta t_1 \mathbf{P} \boldsymbol{\alpha}^1 - r \Delta t_1 \lambda^1 \quad (5.3)$$

$$\mathbf{L}(\boldsymbol{\alpha}^n - \beta_1^n \boldsymbol{\alpha}^{n-1} + \beta_2^n \boldsymbol{\alpha}^{n-2}) + \lambda^n - \beta_1^n \lambda^{n-1} + \beta_2^n \lambda^{n-2} = -\beta \mathbf{P} \boldsymbol{\alpha}^n - r \beta \lambda^n.$$

Para avanzar en el tiempo, es necesario tener un sistema de ecuaciones para  $\boldsymbol{\alpha}^n, \lambda^n$ . Las ecuaciones en (5.3) se pueden reescribir como

$$(\mathbf{L} + \Delta t_1 \mathbf{P}) \boldsymbol{\alpha}^1 + (1 + r \Delta t_1) \lambda^1 = \mathbf{L} \boldsymbol{\alpha}^0 + \lambda^0 \quad (5.4)$$

$$(\mathbf{L} + \beta \mathbf{P}) \boldsymbol{\alpha}^n + (1 + r \beta) \lambda^n = \mathbf{L} (\beta_1^n \boldsymbol{\alpha}^{n-1} - \beta_2^n \boldsymbol{\alpha}^{n-2}) + \beta_1^n \lambda^{n-1} - \beta_2^n \lambda^{n-2}.$$

Ahora se puede usar el algoritmo FGP05 para resolver las ecuaciones (5.4), cuyas matrices correspondientes son

$$\mathbf{P}_0 = \mathbf{L}, \quad (5.5)$$

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

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

Sean  $\mathbf{x}, x_0$  la solución obtenida del algoritmo FGP05 para la matriz  $\mathbf{P}_m$  y lado derecho  $\mathbf{y}$ , que denotamos por  $(\mathbf{x}, x_0) = \text{FGP05}_m(\mathbf{y})$ . Para marchar en el tiempo se procede como sigue:

**Algoritmo:**

**Puntos del preconditionador:** Construir los conjuntos de puntos  $\mathcal{L}^l$  descritos en la sección 4.1.1.

**Construir preconditionador:** Calcular los coeficientes  $\zeta_j^{ln}$  de las funciones cardinales aproximadas  $\tilde{z}_{ln}, l = 1, \dots, N - 1$ , descritas en la sección 4.1.1 tales que:

$$\begin{aligned}\tilde{z}_{ln}(x) &= \sum_{j \in \mathcal{L}^l} \zeta_j^{ln} \mathbf{P}_m(x - x_j) + \text{constante.} \\ \tilde{z}_{ln}(x_j) &= \delta_{ij}, l = 1, \dots, N - 1, \text{ para } m = 0, 1, 2\end{aligned}\tag{5.6}$$

donde  $\delta_{ij}$  es la delta de Kronecker.

**Avanzar en el tiempo:** Una vez calculado el preconditionador, se procede a resolver los sistemas de ecuaciones que corresponden a cada paso del tiempo

$n = 0$

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

$n = 1$

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

$$n = 2, \dots, M$$

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

*Nota:* Para cada  $n$ , se deben incluir las condiciones de frontera mediante las ecuaciones (3.19), (3.20), (3.21), lo cuál no se mencionó anteriormente para facilitar la escritura.

**Nota 5.1.2** *Se debe observar que, al haber impuesto condiciones sobre los coeficientes  $\beta = \beta_0^n$ , sólo se tienen que construir tres conjuntos distintos de funciones cardinales aproximadas  $\tilde{z}_i$ , lo cual resulta en un ahorro de cálculos importante, dado que construir cada una de estas requiere de  $O(Nq^3)$  operaciones.*

**Ejemplo 5.1.1** *En este ejemplo se utilizan valores de los parámetros tomados de [12],  $K = 10, r = 0.05, \sigma = 0.2, T = 0.5$ , para una opción Call. En la variable  $S$  se utilizan  $N$  puntos uniformemente espaciados en  $[0, 30]$ .*

*La figura 5.1 se muestra la comparación de tiempo de ejecución (izquierda) entre el algoritmo en que se invierte la matriz  $\mathbf{L}$ , y el algoritmo iterativo como función de  $N$ . También se muestra error (en una vecindad del strike  $[K/3, 5K/3]$  para  $T = 0$ ) cometido al aproximar la solución del PVIBS correspondiente a cada método (derecha).*

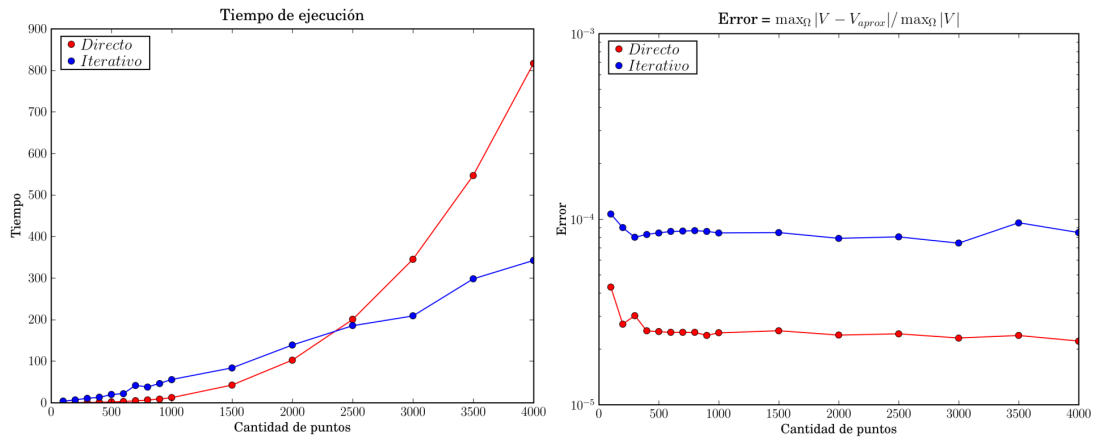


Figura 5.1: Solución de la ecuación (5.1).

La elección de un valor apropiado para el parámetro de forma  $c$  es determinante en la calidad de la aproximación de la solución del PVIBS. Debido a que la teoría de funciones de base radial está apenas en su fase de desarrollo, actualmente no se conoce con precisión qué valores debe tomar para recobrar la mejor aproximación. En [12] se sugiere usar valores cercanos a la menor distancia entre pares de puntos, con el fin de evitar el mal condicionamiento.

Para el ejemplo 5.1 se optimizó el parámetro de forma, es decir, para un número fijo de puntos, se calcula la solución para varios valores de  $c$ , y se elige el valor para el que se obtiene el error más pequeño.

Los resultados mostrados en la figura 5.1 proporcionan evidencia de que al utilizar el algoritmo iterativo FGP05 para resolver los sistemas de ecuaciones correspondientes, se obtiene una reducción importante en los cálculos requeridos para la aproximación de la solución del PVIBS con funciones de base radial.

## 5.2 Suma rápida de multicuádricas con el Método Multipolos

Para el caso unidimensional es posible realizar la traslación de las expansiones de manera un poco más sencilla y barata a la presentada en [32]. En la fórmula (4.19) se observa cómo es posible desacoplar los puntos  $x_i$  e  $y$  en las expansiones, es decir, los coeficientes no dependen del punto de evaluación  $y$ . De esta manera los coeficientes  $Q_l$  no son polinomios, sino constantes para cada  $l$ . Sólo es necesario trasladar el término  $(y - t)^{k-l}$ , es decir, no es necesario calcular  $Q(y)$  en (4.16), con lo que se ahorra la evaluación del producto  $Q_l(\cdot) \|\cdot\|^{2(p+k-l)}$ .

A continuación se dan detalles sobre la traslación de las expansiones correspondientes a los casos  $k = 1, k = -1, k = -3$ , que son las expansiones necesarias para evaluar el operador espacial (5.2).

Sean  $t_1$  el centro original de la expansión (4.19), y  $t_2$  a donde se desea trasladar. Denotemos  $dt = t_2 - t_1$ , para  $k = 1$  la serie centrada en  $t_1$  es

$$r_T(y) = \text{sign}(y - t_1) \sum_{l=0}^{p+1} \frac{Q_l}{(y - t_1)^{l-1}}, \quad (5.7)$$

entonces se desea escribir la suma (5.7) en términos de  $y - t_2$ . Separando la parte polinomial y la parte singular se tiene

$$r_T(y) = \text{sign}(y - t_1) (Q_0(y - t_1) + Q_1) + \text{sign}(y - t_1) \sum_{l=0}^{p-1} \frac{Q_{l+2}}{(y - t_1)^{l+1}},$$

la parte polinomial se traslada fácilmente como sigue

$$\text{sign}(y - t_1) (Q_0(y - t_1) + Q_1) = Q_0(y - t_2) + (Q_0 dt + Q_1),$$

y para la parte singular se utiliza la siguiente fórmula

$$\frac{1}{(y - t_1)^{l+1}} = \sum_{m=l}^{\infty} (-1)^{m-l} \binom{m}{l} dt^{m-l} \frac{1}{y - t_2}. \quad (5.8)$$

La serie trasladada está dada por

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

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

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

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

Similarmente se pueden obtener expresiones para  $k = -1$  y  $k = -3$

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

$$r_T(y) = \text{sign}(y - t_2) \sum_{l=0}^{p-2} \frac{\tilde{Q}_l}{(y - t_2)^{l+3}}, \quad \tilde{Q}_l = \sum_{m=0}^l (-1)^{m-l} \binom{m+2}{l+2} dt^{m-l} Q_l.$$

**Ejemplo 5.2.1** *En este ejemplo se compara el tiempo de ejecución del método multipolos para la evaluación de FBR multicuádricas ( $k = 1$ ) variando la cantidad de puntos  $N$ . Se utilizan puntos aleatoriamente distribuidos en  $[0, 1]$ , se fija el número de términos en las expansiones  $p + k + 1 = 10$  y  $c = 4/N$ .*

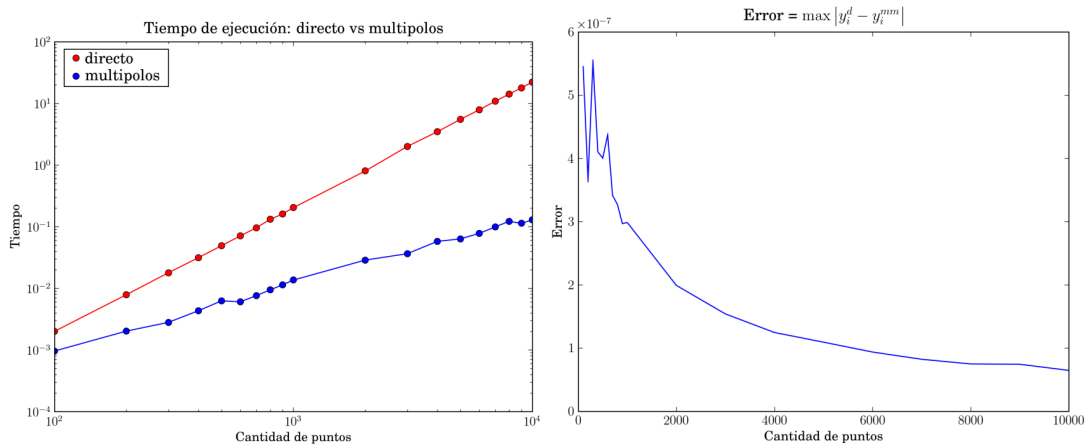


Figura 5.2: Tiempo de ejecución en segundos (izquierda), error (derecha).

### 5.3 Solución del PVIBS utilizando el Método Multipolos

Para la evaluación del operador espacial (5.2) mediante el método multipolos, es necesario reacomodar los términos de la expresión  $r\mathbf{L} - \frac{1}{2}\sigma^2\mathbf{L}_{ss} - r\mathbf{L}_s$  de manera que se puedan ver como una suma similar a la spline (4.12). Denotemos por  $M(\boldsymbol{\lambda}, \mathbf{s}, k)$  la evaluación de la suma (4.11) mediante el método multipolos para FBR multicuádricas, con parámetro  $k$ , coeficientes  $\boldsymbol{\lambda} = [\lambda_i]$  y puntos espaciales  $\mathbf{s} = [s_i]$ . La evaluación de productos  $\mathbf{P}\boldsymbol{\lambda}$  procede como sigue:

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

donde las operaciones  $*$ ,  $(\cdot)^2$  son multiplicación y cuadrado de arreglos elemento a elemento.

**Ejemplo 5.3.1** *En este ejemplo se utiliza el método multipolos para evaluar las multiplicaciones matriz vector necesarias en la solución del PVIBS mediante el algoritmo iterativo FGP05. Los valores de los parámetros son los mismos que los utilizados en el ejemplo 5.1.1. En la figura 5.3 se compara el tiempo de ejecución (izquierda), y el error (derecha) de los distintos métodos.*



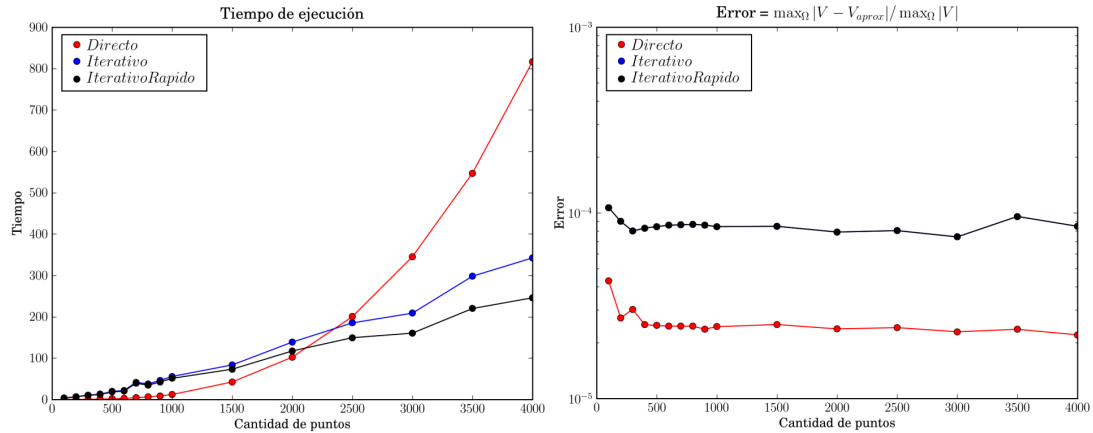


Figura 5.3: directo (rojo), iterativo (azul), iterativo+multipolos (negro).

**Nota 5.3.1** En la figura 5.3 se observa que al incorporar el método multipolos en el algoritmo FGP05, la cantidad de puntos para el cuál comienza a ser eficiente es relativamente alto, con respecto a los resultados mostrados en la figura 5.2. Esto se debe a la forma en que se implementa la multiplicación directa de matrices, esto es, se almacenan en memoria los coeficientes de  $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2$  (los cuales sólo se calculan una vez), de esta manera el método multipolos compite con la evaluación un producto matriz vector estrictamente.

## 5.4 Suma rápida de multicuádras basada en la Transformada Rápida de Gauss Mejorada

Asignar precio a una opción debida a una canasta de bienes es un problema importante (ver sección 2.4), por lo que es interesante resolver el PVIBS en

varias dimensiones. Los métodos para la suma rápida de funciones de base radial basados en la TRGM no son sensibles a la dimensión, por lo que se considera manera importante la posibilidad de aplicar estos métodos en la aproximación de la solución de este problema por medio de funciones de base radial.

Al aproximar la solución del PVIBS por la FBR multicuádrlica  $\sqrt{r^2 + c^2}$ , calcular las derivadas parciales de la función de interpolación  $v(S, t)$  con respecto de la variable espacial  $S$ , implica la evaluación de las FBR multicuádrlicas generalizadas  $(r^2 + c^2)^{-1/2}$  y  $(r^2 + c^2)^{-3/2}$ , por lo cuál es necesario derivar una fórmula similar a (4.20) que sirva para evaluar  $(r^2 + c^2)^{-3/2}$  de manera eficiente. Siguiendo la idea de Roussos para la evaluación rápida de multicuádrlicas, se puede utilizar la fórmula (5.9), que se puede encontrar en [30],

$$(\|x\|^2 + c^2)^{\nu/2} = \frac{1}{\Gamma(\frac{-\nu}{2})} \int_0^\infty \frac{e^{-sc^2}}{s^{\nu/2+1}} e^{-s\|x\|^2} ds \quad (5.9)$$

con parámetro  $\nu < 0$ .

A continuación se ilustra la eficiencia del método de evaluación de FBR multicuádrlicas (4.21) basado en la TRGM de Duraiswami *et al.* Para las pruebas se utilizan puntos uniformemente distribuidos en el hipercubo unitario, para distintas dimensiones. Para todos los ejemplos se fija la precisión pedida a la TRGM como  $10^{-7}$ .

**Ejemplo 5.4.1** *En las figuras 5.4, 5.5 se muestran el tiempo de ejecución del algoritmo rápido comparado con la evaluación directa. Para este ejemplo se fijaron 10 nodos para la evaluación de la cuadratura Gaussiana empleada.*

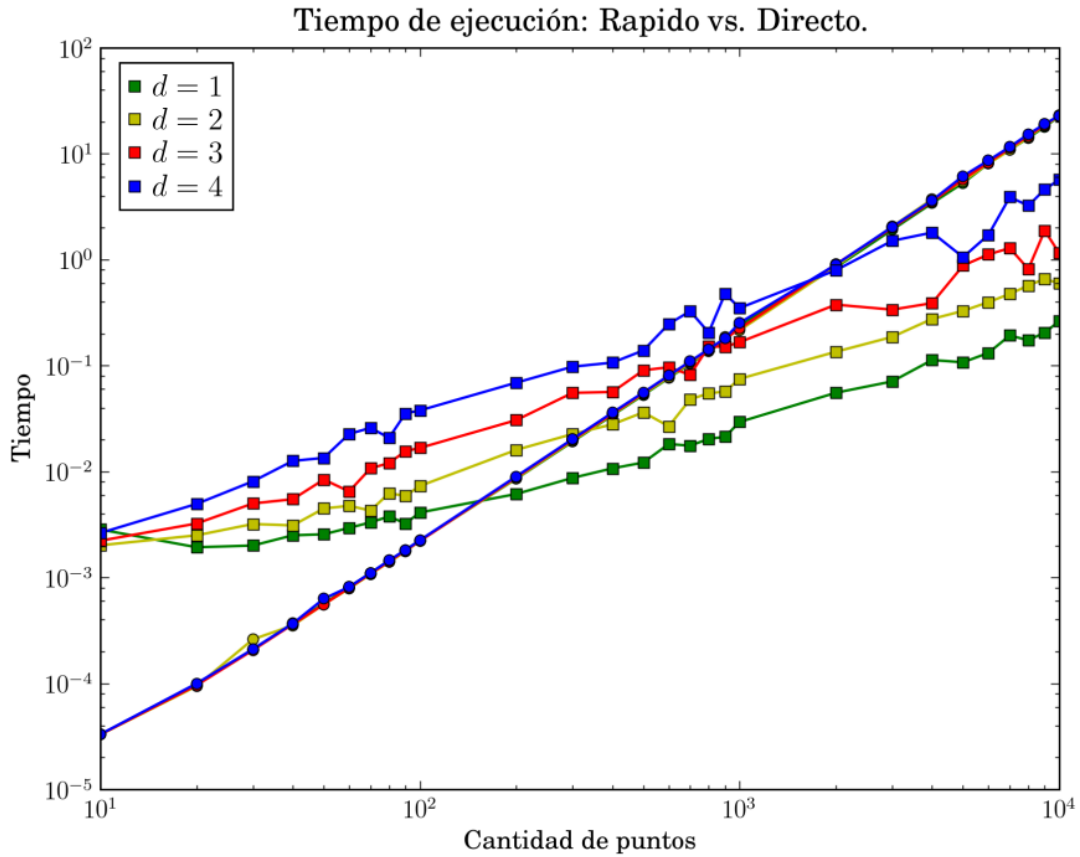


Figura 5.4: Tiempo de ejecución ( $c = 2d$ )  $\circ$  directo,  $\square$  rápido.

Para estos dos ejemplos se obtiene un buen desempeño del algoritmo rápido, y con buena precisión,  $10^{-10}$  aproximadamente para todos los casos. Sin embargo, se observó que con valores más pequeños del parámetro de forma  $c$ , se pierde la precisión rápidamente, resultando en un deterioro de la eficiencia del algoritmo.

**Ejemplo 5.4.2** En la figura 5.6 se muestran el tiempo de ejecución del algoritmo rápido comparado con la evaluación directa. Para este ejemplo se fijaron 100 nodos para la evaluación de la cuadratura Gaussiana empleada (los nece-

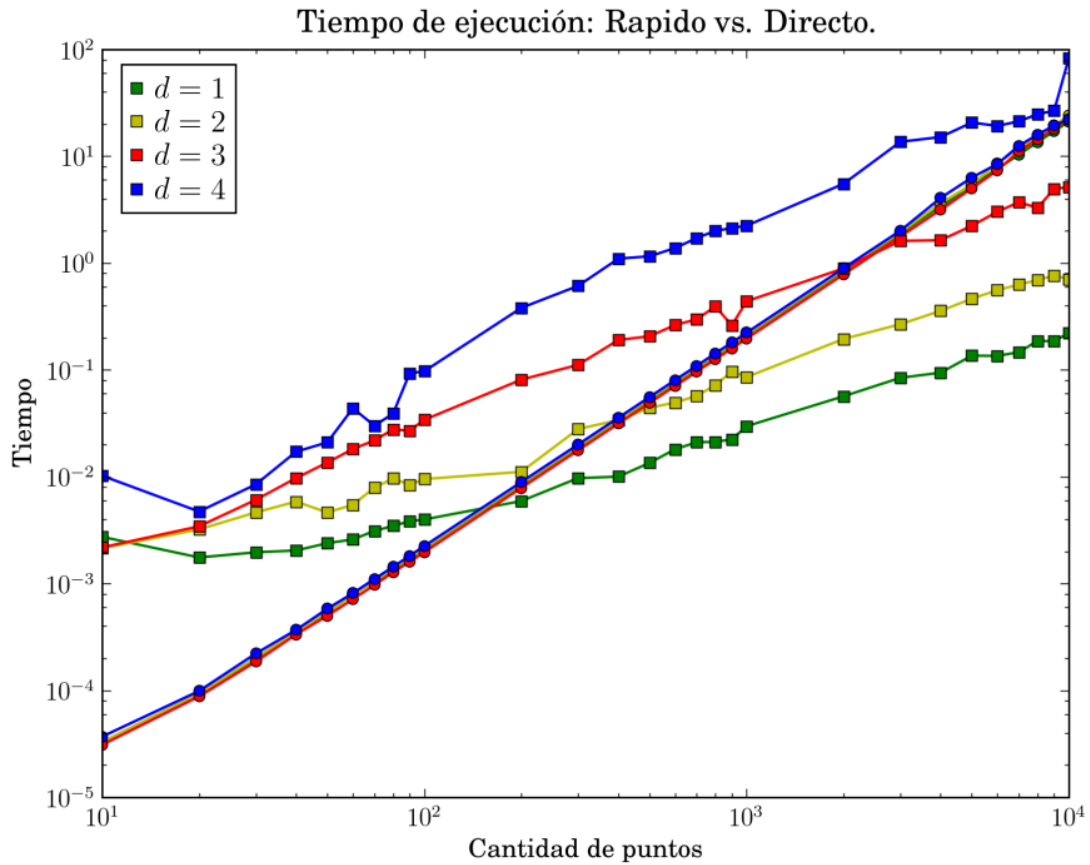


Figura 5.5: Tiempo de ejecución ( $c = 2\sqrt{d}$ )  $\circ$  directo,  $\square$  rápido.

sarios para lograr aproximadamente la misma precisión que en los ejemplos anteriores).

Se realizaron varias pruebas para el método rápido de sumación basado en la TRGM, en las cuales se observó que el desempeño disminuye conforme se toma un valor más pequeño del parámetro de forma, debido a que se requiere una mayor cantidad de evaluaciones de la función (4.23) para lograr la precisión requerida.

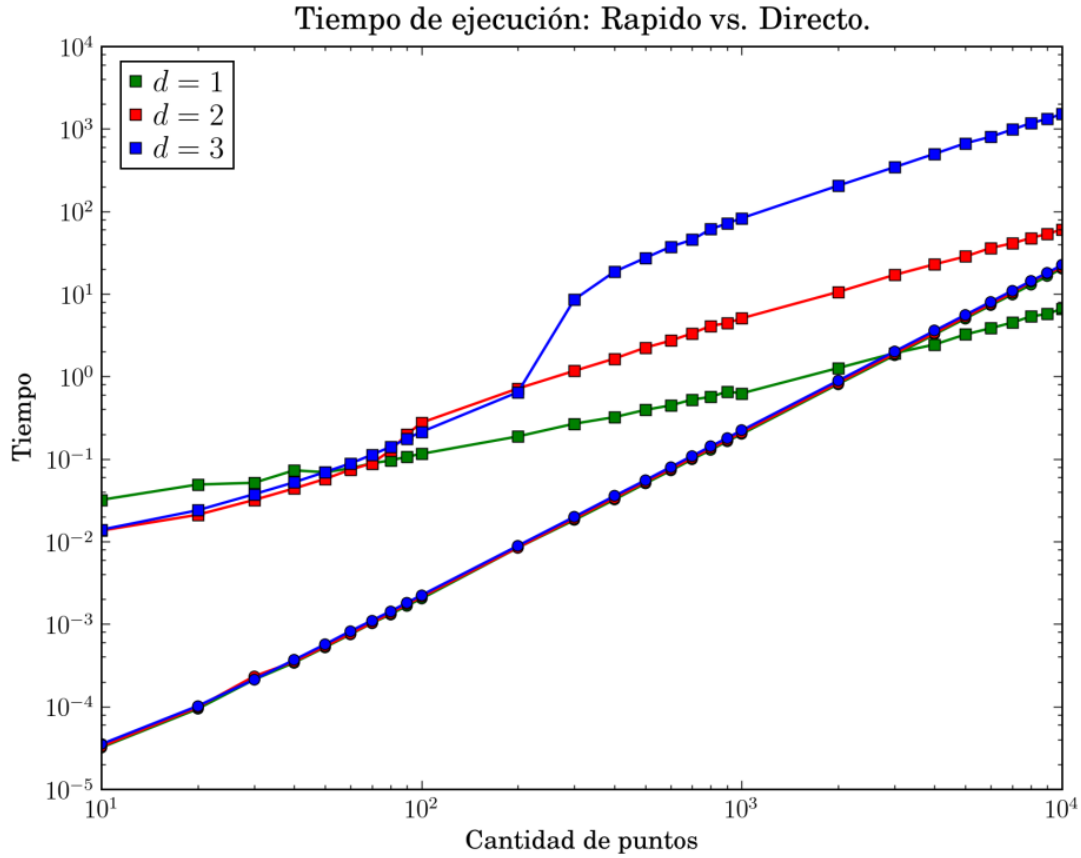


Figura 5.6: Tiempo de ejecución ( $c = 0.2\sqrt{d}$ ) □ rápido, ○ directo.

Para aproximar la solución del PVIJS con FBR multicuádricas se recomienda [12] usar valores del parámetro de forma cercanos a la menor distancia entre los puntos  $x_i$ . Si los puntos estuvieran uniformemente distribuidos, se elige  $c \approx 1/N$ . Debido a esto, no aplicamos este método para evaluar la multiplicación de matrices requeridas en la solución del PVIJS, y queda abierto el problema de mejorar este método de evaluación de FBR multicuádricas para valores pequeños del parámetro de forma.

#### Nota 5.4.1 Detalles sobre las implementaciones

*Para los resultados mostrados en este capítulo se implementó el algoritmo*

*FGP05 y BDF2 en Python [34], utilizando los paquetes de cómputo científico numpy [28], scipy [24] y matplotlib [4].*

*Se implementó una versión recursiva en C++ del método multipolos.*

*La implementación de la TRGM que se utilizó en este trabajo es una versión debida Vlad I. Morariu, originalmente desarrollada en C++ por Vikas C. Raykar y Changjiang Yang, la cual está disponible bajo la licencia LGPL.*

*Para la comunicación desde Python con los algoritmos implementados en C++, se empleó la librería swig [8].*

*Todas las pruebas fueron ejecutadas desde una computadora personal con un procesador AMD64X2 3600+ y 2GB de RAM.*

# Capítulo 6

## Conclusiones

Las pruebas realizadas en la sección 5.1 (figura 5.1) nos muestran evidencia numérica de que es viable el uso de interpolación por medio de funciones de base radial para aproximar la solución del PVIBS.

También se debe notar la utilidad de métodos iterativos como el algoritmo FGP05 (4.1.2) en la solución de los sistemas de ecuaciones lineales que surgen en este problema. En los ejemplos numéricos realizados, se observó un buen desempeño de este algoritmo para resolver sistemas cuyas matrices están dadas por una suma de distintas funciones multicuádricas generalizadas (5.5), en las cuales se incluyen los términos de advección y difusión (para  $\sigma$  y  $r$  constantes), a pesar de que éste fue originalmente para resolver las ecuaciones de interpolación de datos con funciones de base radial multicuádricas.

En la sección 5.3 se muestra el ahorro en el costo computacional obtenido al usar el método multipolos para evaluar los productos matriz vector requeridos en la solución del PVIBS. Se observa un ahorro de tiempo bastante sustancial en los cálculos. Además se debe tomar en cuenta que nuestra implementación del método multipolos es susceptible de ser mejorada en varios aspectos.

No se debe olvidar la importancia de resolver el PVIBS en varias variables (véase sección 2.4), por lo que se debe pensar en algoritmos rápidos para la evaluación de FBR que no sean sensibles a la dimensión. El método de Roussos [18] para la evaluación de FBR multicuádricas parece apropiado. En vista de los resultados numéricos obtenidos en este trabajo, no se aplica este método en la solución del PVIBS, debido a que encontramos cierta compensación entre la eficiencia y el parámetro de forma, por lo que queda como problema abierto resolver esta problemática.

**Nota 6.0.2** *A lo largo de este trabajo nos hemos restringido al uso de FBR multicuádricas, debido a que existen varios trabajos ([26], [29] y [12]) donde se presenta evidencia numérica de su eficiencia al aplicarlas en la solución del PVIBS. Se debe tomar en cuenta de que también existen métodos iterativos y de evaluación rápida para otras funciones de base radial, por ejemplo, las funciones de placa delgada [7],[32], y Gaussianas [5].*



# Bibliografía

- [1] M. J. D. Powell A. C. Faul, G. Goodsell, *A Krylov subspace algorithm for multiquadric interpolation in many dimensions*, IMA Journal of Numerical Analysis (2005), no. 25, 1–24.
- [2] M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Table*, Courier Dover Publications, 1965.
- [3] Y. Achdou and O. Pironneau, *Computational Methods for Option Pricing*, Siam, 2005.
- [4] P. Barrett, J. Hunter, JT Miller, JC Hsu, and P. Greenfield, *matplotlib—A Portable Python Plotting Package*, Astronomical Data Analysis Software and Systems XIV ASP Conference Series, vol. 347, 2005.
- [5] B.J.C. Baxter, *The Interpolation Theory of Radial Basis Functions.*, University of Cambridge, 1992.
- [6] RK Beatson, JB Cherrie, and GN Newsam, *Fast evaluation of radial basis functions: Methods for generalised multiquadrics in  $\mathbb{R}^n$* , SIAM J. Sci. Comput **23** (2002), no. 5, 1549–1571.

- 
- [7] R.K. Beatson and M.J.D. Powell, *An iterative method for thin plate spline interpolation that employs approximations to Lagrange functions*, Numerical Analysis 1993. (D. F. Griffiths, G. A. Watson, eds). (1994), 17–39.
- [8] D.M. Beazley, *SWIG: an easy to use tool for integrating scripting languages with C and C++*, Proceedings of the 4th conference on USENIX Tcl/Tk Workshop, 1996-Volume 4 table of contents, USENIX Association Berkeley, CA, USA, 1996, pp. 15–15.
- [9] F. Black and M. Scholes, *The Pricing of Options and Corporate Liabilities*, Journal of Political Economy **81** (1973), no. 3, 637.
- [10] M.D. Buhmann, *Radial Basis Functions: Theory and Implementations*, Cambridge University Press, 2003.
- [11] R. Duraiswami C. Yang and N.A. Gumerov, *Improved Fast Gauss Transform*, (2003).
- [12] Yiu chung Hon and Xian zhong Mao, *A radial basis function method for solving options pricing model*, Financial Engineering **8** (1999), 31–49.
- [13] B. Lapeyre D. Lamberton, *Introduction to stochastic calculus applied to finance*, 1996.
- [14] S. Arya D. M. Mount, *Library for approximate nearest neighbor searching.*, in Proc. Center for Geometric Computing Second Ann. Fall Workshop Computational Geometry (1997).
- [15] Kansa E.J., *Multiquadrics- A Scattered Data Approximation Scheme with Applications to Computational Fluid Dynamics- II. Solution to Parabolic*,

- Hyperbolic and Elliptic Partial Differential Equations*, Comput. Math. Applic. **19** (1990), 147–161.
- [16] G. Fasshauer, *Meshfree methods*, Handbook of theoretical and computational nanotechnology. Valencia, CA: American Scientific Publishers (2005).
- [17] A. Friedman, *Partial Differential Equations of Parabolic Type.*, Pentice Hall, Englewood Cliffs, NJ, 1964.
- [18] B. J. C. Baxter G. Roussos, *Rapid evaluation of radial basis functions*, J. Comput. Appl. Math. (2005), no. 180, 51–70.
- [19] T. Gonzalez, *Clustering to minimize the maximum intercluster distance*, Theoretical Computer Science (1985), no. 38, 293–306.
- [20] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, Journal of Computational Physics **73** (1987), no. 2, 325–348.
- [21] L. Greengard and J. Strain, *The fast Gauss transform*, SIAM J. SCI. STAT. COMPUT. **12** (1991), no. 1, 79–94.
- [22] N.A. Gumerov and R. Duraiswami, *Fast Radial Basis Function Interpolation via Preconditioned Krylov Iteration*, SIAM Journal on Scientific Computing **29** (2007), no. 5, 1876–1899.
- [23] R.L. Hardy, *Multiquadric equations of topography and other irregular surfaces*, J. Geophys. Res **76** (1971), no. 8, 1905–1915.
- [24] E. Jones, T. Oliphant, P. Peterson, et al., *SciPy: Open source scientific tools for Python*, URL: <http://www.scipy.org> (2001).

- 
- [25] I. Karatzas and S.E. Shreve, *Methods of Mathematical Finance*, Springer Verlag, 1998.
- [26] E. Larsson, K. Åhlander, and A. Hall, *Multi-dimensional option pricing using radial basis functions and the generalized Fourier transform*, Journal of Computational and Applied Mathematics (2007).
- [27] R.C. Merton, *On the pricing of corporate debt: The risk structure of interest rates*, Journal of finance (1974), 449–470.
- [28] T.E. Oliphant and UT Provo, *A Guide to NumPy*, Trelgol Publishing, 2006.
- [29] U. Pettersson, E. Larsson, G. Marcusson, and J. Persson, *Improved radial basis function methods for multi-dimensional option pricing*, Journal of Computational and Applied Mathematics (2007).
- [30] M.J.D. Powell, *The theory of radial basis function approximation in 1990.*, vol. II, Oxford Univ. Press, New York, 1992.
- [31] C. T. Mouat R. K. Beatson, J. B. Cherrie, *Fast fitting of radial basis functions: methods based on preconditioned GMRES iteration*, Advances in Comput. Math. (1999), no. 11, 253–270.
- [32] L. Greengard R.K. Beatson, *A short course on fast multipole methods*, Wavelets, Multilevel Methods and Elliptic PDEs (1997), 1–37.
- [33] H. Greene T. Feder, *Optimal algorithms for approximate clustering*, in Proc. 20th ACM Symp. Theory of Computing, Chicago, Illinois (1988), 434–444.
- [34] G. van Rossum, *Python home page*, URL: <http://www.python.org>.
- [35] P. Wilmott, *Paul Wilmott introduces quantitative finance*, Wiley, 2007.