



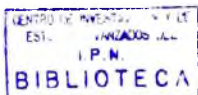
CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS
DEL
INSTITUTO POLITECNICO NACIONAL

DEPARTAMENTO DE INGENIERIA ELECTRICA
SECCION DE COMPUTACION

"SOLUCION SIMBOLICA A LA REPRESENTACION POLINOMIAL DE
OPERADORES GENERALIZADOS DE LIOUVILLE"

Tesis que presenta la Lic. en Ciencias Vicki Anne Saller para
obtener el grado de **MAESTRA EN CIENCIAS** en la especialidad de
Ingeniería Eléctrica. Trabajo dirigido por el Dr. Guillermo
Morales Luna.



CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL
I.P.N.
BIBLIOTECA
INGENIERIA ELECTRICA

México D.F., Septiembre de 1989

*a mi esposo Arturo
por su apoyo y comprensión*

y

*a mi hija Ana Karen
por su infinita paciencia*

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

PREFACIO

La idea para este trabajo surgió de un artículo publicado por el Dr. Jerzy F. Plebański¹ del Departamento de Física del Centro de Investigación y de Estudios Avanzados del IPN. En dicho artículo se habla de la ecuación de Liouville²,

$$\delta x \delta y \phi = e^{-2\epsilon \phi} \quad \text{y} \quad \epsilon^2 = 1, \quad (1)$$

lo cual puede ser expresada en forma equivalente como,

$$F \cdot \delta x \delta y F - \delta x F \cdot \delta y F = \epsilon, \quad \text{donde} \quad F := e^{\epsilon \phi} \quad (2)$$

En términos de \mathcal{L}_1 la ecuación (2) puede escribirse como,

$$\mathcal{L}_1 F = \epsilon, \quad \epsilon^2 = 1 \quad (3)$$

Para $m \geq 2$, \mathcal{L}_m constituye la generalización natural del concepto del "operador de Liouville" \mathcal{L}_1 .

En su trabajo Plebański da soluciones en términos de \mathcal{L}_k para $\mathcal{L}_n(\mathcal{L}_m F)$ donde $n \geq 2$, $m \geq 0$. Basándose en la definición general de \mathcal{L}_k se halla la siguiente expresión general para $\mathcal{L}_n(\mathcal{L}_m F)$,

$$\mathcal{L}_{n+m-1} F \cdot \mathcal{L}_n(\mathcal{L}_m F) = \mathcal{L}_{m-1} F \cdot \mathcal{L}_{n-1}(\mathcal{L}_m F) + \mathcal{L}_{n+m} F + \dots \quad (4)$$

donde se postula que los términos restantes representados por "... " son construcciones algebraicas hechas por $\mathcal{L}_m F$, $m = 0, 1, \dots, k$, $k \leq 2(m+n-1)$. Se expresa interés en determinar la forma explícita

1. Jerzy F. Plebański, *J. Math. Phys.* 29 (1988) 2162.
2. Y. Liouville, *J. Math. Pure Appl.* 18 (1853) 71.

de los términos "...", ya que de esta manera se determinaría el "álgebra" de la composición de los operadores \mathcal{L}_n , $(\mathcal{L}_n \circ \mathcal{L}_m) := \mathcal{L}_n(\mathcal{L}_m F)$.

En los desarrollos algebraicos que conducen a la expansión de $\mathcal{L}_2(\mathcal{L}_1 F)$ se usa la expansión polinomial de $\mathcal{L}_1(FG)$. Es de suponer, entonces, que la expansión polinomial de $\mathcal{L}_2(FG)$ sería necesaria para calcular algebraicamente $\mathcal{L}_3(\mathcal{L}_1 F)$. Plebanski menciona que probablemente no exista tal expansión polinomial a $\mathcal{L}_2(FG)$ y que, por lo tanto, un desarrollo algebraico de $\mathcal{L}_3(\mathcal{L}_1 F)$ sería muy difícil de lograr.

Las generalizaciones de la ecuación de Liouville tienen aplicaciones en el estudio de la relatividad general, así como también, en otras áreas de la física matemática.

En el presente trabajo se busca una expansión general a $\mathcal{L}_n(\mathcal{L}_1 F)$ a través de la computación directa de algunos casos particulares, tales como $\mathcal{L}_3(\mathcal{L}_1 F)$ y $\mathcal{L}_4(\mathcal{L}_1 F)$. De estos resultados se demuestra que no existe una expansión general polinomial para $\mathcal{L}_n(\mathcal{L}_1 F)$ y que los términos "... de la ecuación (4) tampoco coinciden con una representación polinomial, aunque si existen algunos términos polinomiales y otros que son un polinomio multiplicado por un $\partial x^i \partial y^j F$, $i=j$, en la expansión general.

Para completar el trabajo se estudió también la relación $\mathcal{L}_k(FG)$. Por manipulación algebraica y por computación directa se halló el resultado de la operación $\mathcal{L}_2(FG)$ y se vió que, efectivamente, no tiene una representación polinomial en términos de $\mathcal{L}_k F$ y $\mathcal{L}_k G$, $k \geq 2$.

Deseo expresar mi agradecimiento al país de México y en particular al *Centro de Investigación y de Estudios Avanzados del IPN* por haberme brindado la oportunidad de realizar mis estudios de maestría y así también mi gratitud a todos los maestros miembros de la Sección de Computación por su apoyo y estímulo que siempre me ofrecieron, en especial al Dr. Guillermo Morales Luna, quien con su valiosa asesoría y útiles consejos me guió durante la realización de este trabajo.

Deseo expresar también mi más sincero agradecimiento a mis compañeros de la Sección de Computación quienes con su amistad y ayuda hicieron de estos estudios de maestría una época memorable y provechoso en mi vida.

CENTRO DE INVESTIGACIONES Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

CONTENIDO

1. INTRODUCCION	1
2. BANCO DE FORMULAS CONOCIDAS	4
3. CALCULO ALGEBRAICO DE $\mathcal{L}_n(FG)$	8
4. CALCULO DE $\mathcal{L}_n(\mathcal{L}_m F)$ VIA LA COMPUTACION DIRECTA	15
5. CALCULO DE $\mathcal{L}_n(FG)$ VIA LA COMPUTACION DIRECTA	34
6. RESULTADO SIMPLIFICADO DE $\mathcal{L}_3(\mathcal{L}_1 F)$	37
7. CARACTERISTICAS DE LA EXPANSION DE $\mathcal{L}_n(\mathcal{L}_m F)$	43
8. SIMPLIFICACION PARCIAL A $\mathcal{L}_n(\mathcal{L}_1 F)$	51
9. CARACTERISTICAS DE LA EXPANSION DE $\mathcal{L}_n(FG)$	55
10. CONCLUSIONES	57

Capítulo 1. Introducción.

Sea \mathcal{F} el conjunto de funciones infinitamente diferenciables de dos variables x, y . Considérese entonces la sucesión de transformaciones no-lineales diferenciales $\mathcal{L}_m: \mathcal{F} \rightarrow \mathcal{F}$, $k = 1, 2, \dots$ definidas por:

$$F \in \mathcal{F} \Rightarrow \mathcal{L}_m F := \det(\partial x^i \partial y^k F), \quad i, k = 0, 1, \dots, m, \quad m > 0 \quad (1.1)$$

donde la entrada general F_{ik} se interpreta como $\partial x^i \partial y^k F$, i y k siendo las iteraciones de derivadas parciales en x y en y , respectivamente, sobre F y donde $\partial x^0 = \partial y^0 = 1$.

Esta definición puede ser extendida a incluir a todo entero m , postulando que,

$$\begin{aligned} m = 0 &\implies \mathcal{L}_m F = F \\ m = -1 &\implies \mathcal{L}_m F = 1 \\ m < -2 &\implies \mathcal{L}_m F = 0 \end{aligned} \quad (1.2)$$

En la publicación del Dr. Plebański se demuestran las siguientes identidades básicas del operador \mathcal{L}_m ,

$$\mathcal{L}_1(\mathcal{L}_m F) = \mathcal{L}_{m-1} F \cdot \mathcal{L}_{m+1} F \quad (1.3)$$

$$\mathcal{L}_2(\mathcal{L}_m F) = (\mathcal{L}_{m-1} F)^2 \cdot \mathcal{L}_{m+2} F + (\mathcal{L}_{m+1} F)^2 \cdot \mathcal{L}_{m-2} F \quad (1.4)$$

El problema que se plantea es la generalización de estas fórmulas a una representación polinomial de $\mathcal{L}_n(\mathcal{L}_m F)$ escrito como

$$\mathcal{L}_{n+m-1} F \cdot \mathcal{L}_n(\mathcal{L}_m F) = \mathcal{L}_{m-1} F \cdot \mathcal{L}_{n-1}(\mathcal{L}_m F) \cdot \mathcal{L}_{n+m} F + \dots \quad (1.5)$$

donde basta con buscar una expansión para $\mathcal{L}_n(\mathcal{L}_m F)$ y luego hacer la multiplicación por $\mathcal{L}_{n+m-1} F$ para que quede en forma polinomial.

Para llegar a las expansiones de $\mathcal{L}_2(\mathcal{L}_m F)$ algebraicamente se usó la expansión de $\mathcal{L}_1(FG)$,

$$\mathcal{L}_1(FG) = F^2 \mathcal{L}_1 G + G^2 \mathcal{L}_1 F \quad (1.6)$$

Por simetría con el caso de $\mathcal{L}_2(\mathcal{L}_m F)$, Plebański conjetura que la expansión de $\mathcal{L}_2(\mathcal{L}_m F)$ requiere de la expansión de $\mathcal{L}_2(FG)$ para su cálculo algebraico. El comenta que, debido a esta dependencia, un cálculo algebraico de $\mathcal{L}_n(\mathcal{L}_m F)$, $n > 2$, ha de ser muy difícil, ya que parece ser que $\mathcal{L}_2(FG)$ no tiene una representación polinomial en términos del operador \mathcal{L}_k . El sugiere que se hagan unos cálculos por vía directa de la computación de $\mathcal{L}_n(\mathcal{L}_m F)$, $n > 2$, para conocer la naturaleza de estos resultados.

El objetivo de esta tesis es analizar la expansión general a $\mathcal{L}_n(\mathcal{L}_m F)$ por vía directa de la computación para verificar si existe una expansión polinomial general directa o en forma de la ecuación (1.5).

Para llegar a una solución del problema de $\mathcal{L}_n(\mathcal{L}_m F)$, lo primero que se hizo fue resumir en un banco todas las ecuaciones conocidas para $\mathcal{L}_n(\mathcal{L}_m F)$, $n < 3$, y $\mathcal{L}_m(FG)$, $m < 2$, con la esperanza de encontrar alguna fórmula algebraica para el caso de $\mathcal{L}_3(\mathcal{L}_m F)$. Tal fórmula no se halló, pero sí sirvieron estas ecuaciones del banco para calcular algebraicamente a $\mathcal{L}_2(FG)$. Se vió que, en efecto, la representación para $\mathcal{L}_2(FG)$ no es un polinomio en términos de \mathcal{L}_k 's. Esta representación se corroboró con un cálculo directo vía computadora.

Después, se hizo un programa que calcula $\mathcal{L}_n(\mathcal{L}_m F)$ directamente con las restricciones sobre n y m dependientes del tiempo y cantidad de memoria disponibles. El estilo del programa siguió al que se usó para calcular a $\mathcal{L}_2(FG)$. Se comprobó su buen funcionamiento al hacerlo reproducir las representaciones polinomiales para $\mathcal{L}_n(\mathcal{L}_m F)$, $n < 3$.

Con este programa se hizo la expansión completa de $\mathcal{L}_3(\mathcal{L}_1 F)$ como una suma de términos. El programa se encargó de eliminar términos que se cancelan entre si y de registrar el número de veces que aparecen los términos no cancelables. La simplificación de grupos de términos a $\mathcal{L}_k F$'s se hizo también por

computadora, aunque fue necesario hacer un reagrupación manual de los términos para llegar a una simplificación completa.

Del resultado para el caso de $\mathcal{L}_3(\mathcal{L}_1 F)$ se pudo sacar conclusiones acerca de las características de una respuesta general a $\mathcal{L}_n(\mathcal{L}_n F)$. La conclusión principal fue que para $n > 2$ no hay una representación polinomial completa en términos de los $\mathcal{L}_k F$'s, ni aun haciendo la multiplicación sugerida en la ecuación (1.5). Se demostró el porqué, basándose en las características generales de los términos que conforman las respuestas a las fórmulas $\mathcal{L}_n(\mathcal{L}_n F)$. Sin embargo, en el capítulo 8 se dan unas fórmulas que sugieran que la respuesta general a $\mathcal{L}_n(\mathcal{L}_n F)$ siempre contendrá un término polinomial, más algunos otros términos de forma polinomial multiplicados por factores regulares del tipo $\partial x^i \partial y^j F$, donde $i=j$.

Capítulo 2. Banco de Fórmulas Conocidas.

En este capítulo se da un repaso sobre las ecuaciones mencionadas por el Dr. Plebanski sobre el manejo algebraico de las operaciones $\mathcal{L}_n(\mathcal{L}F)$ y $\mathcal{L}_m(FG)$. Se da una relación para \mathcal{L}_1F en términos del logaritmo natural, lo cual sirve para calcular el resultado a $\mathcal{L}_1(FG)$. A la vez, se desarrolla una relación para $\mathcal{L}_1(F^a)$ que se necesita en el capítulo 4 para el cálculo algebraico de $\mathcal{L}_2(FG)$. Luego se nota la fuerte dependencia que hay entre las operaciones $\mathcal{L}_1(FG)$ y $\mathcal{L}_2(\mathcal{L}F)$ al ver cómo se necesita el resultado de $\mathcal{L}_1(FG)$ para calcular $\mathcal{L}_2(\mathcal{L}F)$. Finalmente, se desarrolla un resultado parcial para la operación general $\mathcal{L}_n(\mathcal{L}F)$.

La primera relación interesante por mostrar es que \mathcal{L}_1F puede expresarse como $F^2\delta x\delta y(\ln(F))$. Recordando que,

$$\frac{d}{dx} \ln(x) = \frac{1}{x}$$

y que

$$\frac{d}{dx} f(g(x)) = \frac{d}{dx} f(g(x)) \cdot \frac{dg}{dx}$$

entonces,

$$\delta y \ln(F) = \frac{1}{F} \delta y F = F^{-1} \delta y F$$

de donde se tiene que,

$$\delta x \delta y \ln(F) = \delta x (F^{-1} \delta y F) = -F^{-2} \delta x F \delta y F + F^{-1} \delta x \delta y F \quad (2.1)$$

Multiplicando ambos lados de la ecuación (2.1) por F^2 da

$$F^2 \delta x \delta y \ln(F) = F \delta x \delta y F - \delta x F \delta y F = \left| \begin{array}{c} F \\ \delta y F \end{array} \frac{\delta x F}{\delta x \delta y F} \right| = \mathcal{L}_1 F \quad (2.2)$$

Una relación que se usa para solucionar $\mathcal{L}_2(FG)$ se halla basándose en la ecuación (2.2),

$$\begin{aligned} \mathcal{L}_1(F^\alpha) &= F^{2\alpha} \partial x \partial y (\ln(F^\alpha)) = \alpha F^{2\alpha} \partial x \partial y (\ln(F)) = \\ &= \alpha F^{2(\alpha-1)} F^2 \partial x \partial y (\ln(F)) = \alpha F^{2(\alpha-1)} \mathcal{L}_1 F \end{aligned} \quad (2.3)$$

Sustituyendo FG en lugar de F en la ecuación (2.2) se puede demostrar a la ecuación (1.6),

$$\begin{aligned} \mathcal{L}_1(FG) &= (FG)^2 \partial x \partial y \ln(FG) = (FG)^2 \partial x \partial y (\ln(F) + \ln(G)) \\ &= (FG)^2 F^{-2} \mathcal{L}_1 F + (FG)^2 G^{-2} \mathcal{L}_1 G \\ &= G^2 \mathcal{L}_1 F + F^2 \mathcal{L}_1 G \end{aligned} \quad (2.4)$$

La comprobación algebraica de la ecuación (1.3) no es trivial, pero la ecuación (1.4) sí se puede demostrar fácilmente utilizando a las ecuaciones (1.3) y (2.4). En efecto, tenemos primeramente que,

$$\mathcal{L}_1(\mathcal{L}_1(\mathcal{L}_m F)) = \mathcal{L}_1(\mathcal{L}_{m-1} F \cdot \mathcal{L}_{m+1} F) \quad (\text{por ec. 1.3})$$

Ahora, se tiene por la ecuación (2.4) que

$$\mathcal{L}_1(\mathcal{L}_1(\mathcal{L}_m F)) = (\mathcal{L}_{m+1} F)^2 \mathcal{L}_1(\mathcal{L}_{m-1} F) + (\mathcal{L}_{m-1} F)^2 \mathcal{L}_1(\mathcal{L}_{m+1} F) \quad (2.5)$$

Desarrollando el lado izquierdo de la ecuación 2.5 se tiene por la ecuación (1.3) que,

$$\mathcal{L}_1(\mathcal{L}_1(\mathcal{L}_m F)) = \mathcal{L}_m F \cdot \mathcal{L}_2(\mathcal{L}_m F) \quad (\text{izq})$$

y desarrollando el lado derecho de la ecuación (2.5) se tiene por la ecuación (2.4) que,

$$\begin{aligned} (\mathcal{L}_{m+1} F)^2 \mathcal{L}_1(\mathcal{L}_{m-1} F) + (\mathcal{L}_{m-1} F)^2 \mathcal{L}_1(\mathcal{L}_{m+1} F) &= \\ = (\mathcal{L}_{m+1} F)^2 \mathcal{L}_{m-2} F \cdot \mathcal{L}_m F + (\mathcal{L}_{m-1} F)^2 \mathcal{L}_{m+2} F \cdot \mathcal{L}_m F \end{aligned} \quad (\text{der})$$

Cancelando $\mathcal{L}_m F$ a ambos lados, da como resultado que,

$$\mathcal{L}_2(\mathcal{L}_m F) = (\mathcal{L}_{m+1} F)^2 \mathcal{L}_{m-2} F + (\mathcal{L}_{m-1} F)^2 \mathcal{L}_{m+2} F \quad (2.6)$$

Se notará que para hallar $\mathcal{L}_2(\mathcal{L}_m F)$ se usó el resultado de $\mathcal{L}_1(FG)$. Se esperaría, en consecuencia, que el cálculo algebraico de $\mathcal{L}_3(\mathcal{L}_m F)$ depende del resultado de $\mathcal{L}_2(FG)$.

Las últimas fórmulas que quedan por describir dan una respuesta parcial a $\mathcal{L}_n(\mathcal{L}_m F)$ en su forma general donde no hay restricción sobre el valor de n. Por definición se tiene que,

$$\mathcal{L}_m F = \mathcal{L}_{m-1} F \cdot \partial x^m \partial y^m F + \dots \quad (2.7)$$

donde "... " son los $m \cdot m!$ términos restantes de los $(m + 1)!$ términos de $\mathcal{L}_m F$. Aquí y en lo sucesivo, un término es un sumando tomado por un producto de factores de la forma $\partial x^i \partial y^j F$. Luego, basándose en la ecuación 2.7, resulta que,

$$\begin{aligned} \mathcal{L}_n(\mathcal{L}_m F) &= \mathcal{L}_{n-1}(\mathcal{L}_m F) \cdot \partial x^n \partial y^n(\mathcal{L}_m F) + \dots \\ &= \mathcal{L}_{n-1}(\mathcal{L}_m F) \cdot \partial x^n \partial y^n((\mathcal{L}_{m-1} F) \cdot \partial x^m \partial y^m F) + \dots \\ &= \mathcal{L}_{m-1} F \cdot \mathcal{L}_{n-1}(\mathcal{L}_m F) \cdot \partial x^{n+m} \partial y^{n+m} F + \dots \end{aligned}$$

(2.8)

En la ecuación 2.8 "... " representa a todos los términos restantes de la fórmula $\mathcal{L}_n(\mathcal{L}_m F)$. En el caso particular de $n = 3$, $m = 1$, hay 1200 términos distinguibles en la expansión completa. Lo que se quiere decir con términos "distinguibles" es que son distintos unos de otros al comparar los valores de los índices de sus factores. La cuenta de términos en total toma en cuenta el número de repeticiones de cada término distinguible. Se comenta esto para hacer notar que no es fácil averiguar el contenido de "...".

Para el caso de $n = 3$, el resultado parcial expresado por la ecuación 2.8 cae dentro del rango previamente definido.

$$\mathcal{L}_{n-1} F \cdot \mathcal{L}_2 (\mathcal{L}_n F) \cdot \delta x^{n+m} \delta y^{n+m} \quad (2.9)$$

Entonces, la ecuación 2.9 sirve como una manera de verificar en parte si la expansión propuesta para $\mathcal{L}_3 (\mathcal{L}_n F)$, con una n dado, es correcta. Se verá en el capítulo 6 que la ecuación 2.9 sí se cumple en las expansiones halladas por vía directa de la computación para los casos $n = 1$ y $n = 2$ donde $m = 3$.

Al multiplicar la ecuación (2.8) por $\mathcal{L}_{n+m-1} F$ y empleando la ecuación (2.7) se consigue una ecuación polinomial en términos del operador \mathcal{L}_k para la fórmula general $\mathcal{L}_n (\mathcal{L}_n F)$,

$$\mathcal{L}_{n+m-1} F \cdot \mathcal{L}_n (\mathcal{L}_n F) = \mathcal{L}_{n-1} F \cdot \mathcal{L}_{n-1} (\mathcal{L}_n F) \cdot \mathcal{L}_{n+m} F + \dots \quad (2.10)$$

La ecuación (2.10) es la que propone Plebanski que pueda dar una expansión general polinomial para la iteración $\mathcal{L}_n (\mathcal{L}_n F)$, ya que la ecuación (2.8) no sirve debido a la presencia del factor $\delta x^{n+m} \delta y^{n+m} F$.

Capítulo 3. Cálculo Algebraico de $\mathcal{L}_2(FG)$.

Usando algunas de las ecuaciones mencionadas en el capítulo 1 y otras de las desarrolladas en el capítulo 2 se muestra en este capítulo un cálculo algebraico del resultado de $\mathcal{L}_2(FG)$. Se verá que dicho resultado no está en forma polinomial, sino que la mayoría de los términos tienen factores residuales que no son reducibles a conjuntos de $\mathcal{L}_k F$ y $\mathcal{L}_k G$. En el capítulo 9 se discute las características generales de $\mathcal{L}_k(FG)$ y se demuestra que no hay manera de desaparecer a los residuos en los resultados para $\mathcal{L}_k(FG)$, $k > 1$.

Usando ec. 1.3 tenemos que $\mathcal{L}_1(\mathcal{L}_1(FG)) = \mathcal{L}_0(FG) \cdot \mathcal{L}_2(FG)$, de donde se deduce que,

$$\mathcal{L}_2(FG) = \frac{1}{FG} \mathcal{L}_1(\mathcal{L}_1(FG)) \quad (3.1)$$

donde, aplicando la ecuación 1.6, da

$$\mathcal{L}_2(FG) = \frac{1}{FG} \mathcal{L}_1(G^2 \mathcal{L}_1 F + F^2 \mathcal{L}_1 G) \quad (3.2)$$

Para determinar si $\mathcal{L}_2(FG)$ es representable como un polinomio en términos de \mathcal{L}_k 's hay que desarrollar la ecuación (2.2) hasta eliminar la división por FG. Haciendo algunas substitutiones para simplificar la representación, digamos, $a = G^2 \mathcal{L}_1 F$ y $b = F^2 \mathcal{L}_1 G$, se tiene que,

$$\mathcal{L}_1(a + b) =$$

$$\begin{aligned} & \begin{vmatrix} (a+b)_{00} & (a+b)_{01} \\ (a+b)_{10} & (a+b)_{11} \end{vmatrix} = \begin{vmatrix} a_{00} + b_{00} & a_{01} + b_{01} \\ a_{10} + b_{10} & a_{11} + b_{11} \end{vmatrix} \\ & = \begin{vmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{vmatrix} + \begin{vmatrix} a_{00} & b_{01} \\ a_{10} & b_{11} \end{vmatrix} + \begin{vmatrix} b_{00} & a_{01} \\ b_{10} & a_{11} \end{vmatrix} + \begin{vmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{vmatrix} \\ & = \mathcal{L}_1 a + \mathcal{L}_1 b + \begin{vmatrix} a_{00} & b_{01} \\ a_{10} & b_{11} \end{vmatrix}_{(\det_1)} + \begin{vmatrix} b_{00} & a_{01} \\ b_{10} & a_{11} \end{vmatrix}_{(\det_2)} \end{aligned}$$

de donde se puede escribir la siguiente representación de $\mathcal{L}_2(FG)$,

$$\mathcal{L}_2(FG) = \frac{1}{FG} \left(\mathcal{L}_1(F^2 \mathcal{L}_1 G) + \mathcal{L}_1(G^2 \mathcal{L}_1 F) \right) + \frac{1}{FG} (\det_1 + \det_2)$$

(3.3)

Resolviendo la ecuación (3.3) por partes, se tiene primero que,

$$\begin{aligned} \mathcal{L}_1(F^2 \mathcal{L}_1 G) &= (\mathcal{L}_1 G)^2 \mathcal{L}_1(F^2) + F^4 \mathcal{L}_1(\mathcal{L}_1 G) && \text{(por ec. 1.6)} \\ &= (\mathcal{L}_1 G)^2 \mathcal{L}_1(F^2) + F^4 G \cdot \mathcal{L}_2 G && \text{(por ec. 1.3)} \\ &= (\mathcal{L}_1 G)^2 (2F^2 \mathcal{L}_1 F) + F^4 G \cdot \mathcal{L}_2 G && \text{(por ec. 2.3)} \\ &= F^2 (2(\mathcal{L}_1 G)^2 \mathcal{L}_1 F + F^2 G \cdot \mathcal{L}_2 G) \end{aligned}$$

de donde se tiene que,

$$\begin{aligned}
& \frac{1}{FG} \left(\mathcal{L}_1(F^2 \mathcal{L}_1 G) + \mathcal{L}_1(G^2 \mathcal{L}_1 F) \right) = \\
& = \frac{1}{FG} \left[F^2 (2(\mathcal{L}_1 G)^2 \mathcal{L}_1 F + F^2 G \cdot \mathcal{L}_2 G) + G^2 (2(\mathcal{L}_1 F)^2 \mathcal{L}_1 G + G^2 F \cdot \mathcal{L}_2 F) \right] \\
& = \frac{2}{FG} \left(F^2 (\mathcal{L}_1 G)^2 \mathcal{L}_1 F + G^2 (\mathcal{L}_1 F)^2 \mathcal{L}_1 G \right) + F^3 \mathcal{L}_2 G + G^3 \mathcal{L}_2 F
\end{aligned}
\tag{3.4}$$

Falta desarrollar los \det_1 y \det_2 . Basta con desarrollar el \det_1 para conocer el \det_2 ya que son simétricos al cambiar los F por los G y los G por los F. Se tiene que el \det_1 es igual a $(a_{00} b_{11} - a_{10} b_{01})$.

Cálculo de \det_1 :

$$\begin{aligned}
a_{00} b_{11} &= G^2 \mathcal{L}_1 F \cdot (F^2 \mathcal{L}_1 G)_{,11} = G^2 \mathcal{L}_1 F \cdot \partial x \partial y (F^2 (G_{00} G_{11} - G_{10} G_{01})) = \\
&= G^2 (F_{00} F_{11} - F_{10} F_{01}) \cdot \left[2(F_{00} F_{11} G_{00} G_{11} + F_{01} F_{10} G_{00} G_{11} \right. \\
&\quad + F_{00} F_{10} G_{00} G_{12} + F_{00} F_{01} G_{00} G_{21} \\
&\quad - F_{00} F_{11} G_{01} G_{10} - F_{01} F_{10} G_{10} G_{01} \\
&\quad - F_{00} F_{10} G_{02} G_{10} - F_{00} F_{01} G_{01} G_{20}) \\
&\quad \left. + F_{00} F_{00} G_{00} G_{22} - F_{00} F_{00} G_{02} G_{20} \right] \\
&= G^2 \left[2 \mathcal{L}_1 G ((F_{00} F_{11})^2 - (F_{01} F_{10})^2) + F^2 \mathcal{L}_1 F (G_{00} G_{22} - G_{02} G_{20}) \right. \\
&\quad \left. + F \mathcal{L}_1 F (F_{10} (G_{00} G_{12} - G_{02} G_{10}) + F_{01} (G_{00} G_{21} - G_{01} G_{20})) \right]
\end{aligned}
\tag{3.5}$$

Ahora, para calcular $a_{01} b_{10}$ basta con conocer a_{01} , ya que b_{10} se conocerá entonces por simetría, i.e.,

$$\begin{aligned}
a_{01} &= G^2(F_{00}F_{11} - F_{01}F_{10})_{01} = 2(G_{00}G_{01}F_{00}F_{11} - G_{00}G_{01}F_{01}F_{10}) \\
&+ G_{00}G_{00}F_{01}F_{11} + G_{00}G_{00}F_{00}F_{12} - G_{00}G_{00}F_{02}F_{10} - G_{00}G_{00}F_{01}F_{11} \\
&= 2G_{00}G_{01}\mathcal{L}_1F + G^2(F_{00}F_{12} - F_{02}F_{10}),
\end{aligned}$$

de donde se tiene que,

$$\begin{aligned}
a_{01}b_{10} &= 2G_{00}G_{01}\mathcal{L}_1F(F^2(G_{00}G_{21} - G_{01}G_{20}) + 2F_{00}F_{10}\mathcal{L}_1G) \\
&+ G^2(F_{00}F_{12} - F_{02}F_{10})(2F_{00}F_{10}\mathcal{L}_1G + F^2(G_{00}G_{21} - G_{01}G_{20}))
\end{aligned}$$

(3.6)

Resumiendo, el \det_1 es igual a la ecuación (3.5) menos la ecuación (3.6) y el \det_2 es igual a \det_1 , excepto por un intercambio de todos los F's por G's y todos los G's por F's.

Juntando las ecuaciones (3.4), (3.5), y (3.6) y haciendo las divisiones necesarias por FG en las ecuaciones (3.5) y (3.6), se obtiene la siguiente representación de $\mathcal{L}_2(FG)$,

$$\begin{aligned}
\mathcal{L}_2(FG) &= F^3 \mathcal{L}_2 G + G^3 \mathcal{L}_2 F + \frac{2}{FG} \left(F^3 \mathcal{L}_1 F \left((G_{00} G_{11})^2 + (G_{01} G_{10})^2 \right. \right. \\
&- \left. \left. 2G_{00} G_{11} G_{01} G_{10} \right) + G^3 \mathcal{L}_1 G \left((F_{00} F_{11})^2 + (F_{01} F_{10})^2 - 2F_{00} F_{11} F_{01} F_{10} \right) \right) \\
&+ \frac{G^2}{FG} \left(2\mathcal{L}_1 G \left((F_{00} F_{11})^2 - (F_{01} F_{10})^2 \right) + 2F \mathcal{L}_1 F (G_{10} (G_{00} G_{12} - G_{02} G_{10}) \right. \\
&\quad \left. + F_{01} (G_{00} G_{21} - G_{01} G_{20})) + F^2 \mathcal{L}_1 F (G_{00} G_{22} - G_{02} G_{20}) \right) \\
&+ \frac{F^2}{FG} \left(2\mathcal{L}_1 F \left((G_{00} G_{11})^2 - (G_{01} G_{10})^2 \right) + 2G \mathcal{L}_1 G (G_{10} (F_{00} F_{12} - F_{02} F_{10}) \right. \\
&\quad \left. + G_{01} (F_{00} F_{21} - F_{01} F_{20})) + G^2 \mathcal{L}_1 G (F_{00} F_{22} - F_{02} F_{20}) \right) \\
&- \frac{2}{FG} G_{00} G_{01} \mathcal{L}_1 F (F^2 (G_{00} G_{21} - G_{01} G_{20}) + 2F_{00} F_{10} \mathcal{L}_1 G) \\
&- \frac{2}{FG} F_{00} F_{01} \mathcal{L}_1 G (G^2 (F_{00} F_{21} - F_{01} F_{20}) + 2G_{00} G_{10} \mathcal{L}_1 F) \\
&- \frac{G^2}{FG} (F_{00} F_{12} - F_{02} F_{10}) (2F_{00} F_{10} \mathcal{L}_1 G + F^2 (G_{00} G_{21} - G_{01} G_{20})) \\
&- \frac{F^2}{FG} (G_{00} G_{12} - G_{02} G_{10}) (2G_{00} G_{10} \mathcal{L}_1 F + G^2 (F_{00} F_{21} - F_{01} F_{20}))
\end{aligned}$$

(3.7)

Eliminando los términos que se cancelan entre sí en la ecuación (3.7), todos los términos restantes son divisibles por FG, (los términos que se cancelan se subrayan.) Reacomodando los términos restantes se llega a la siguiente representación simplificada para $\mathcal{L}_2(FG)$, (se indica parte de la simetría presente):

Resultado Simplificado de $L_2(FG)$:

$$\begin{aligned}
 \mathcal{L}_2(FG) = & \overbrace{F^3 \mathcal{L}_2 G + G^3 \mathcal{L}_2 F} - F^2 G^2 (F_{12} G_{21} + F_{21} G_{12}) \\
 & + 4 \mathcal{L}_1 F \mathcal{L}_1 G (F_{00} G_{11} + F_{11} G_{00} - F_{01} G_{10} - F_{10} G_{01}) \\
 - 2 FG \mathcal{L}_1 F (G_{01} G_{21} + G_{10} G_{12}) & - 2 FG \mathcal{L}_1 G (F_{01} F_{21} + F_{10} F_{12}) \\
 + FG \mathcal{L}_1 F (G_{00} G_{22} - G_{02} G_{20}) & + FG \mathcal{L}_1 G (F_{00} F_{22} - F_{02} F_{20}) \\
 + 2 G^2 \mathcal{L}_1 F (F_{01} G_{21} + F_{10} G_{12}) & + 2 F^2 \mathcal{L}_1 G (F_{21} G_{01} + F_{12} G_{10}) \\
 + 2 F \mathcal{L}_1 F (G_{01} G_{01} G_{20} + G_{02} G_{10} G_{10}) & \\
 + 2 G \mathcal{L}_1 G (F_{01} F_{01} F_{20} + F_{02} F_{10} F_{10}) & \\
 - 2 G \mathcal{L}_1 F (F_{01} G_{01} G_{20} + F_{10} G_{02} G_{10}) & \\
 - 2 F \mathcal{L}_1 G (F_{01} F_{20} G_{01} + F_{02} F_{10} G_{10}) & \\
 + F^2 G (F_{12} G_{01} G_{20} + F_{21} G_{02} G_{10}) & + FG^2 (F_{01} F_{20} G_{12} + F_{02} F_{10} G_{21}) \\
 - FG (F_{02} F_{10} G_{01} G_{20} + F_{01} F_{20} G_{02} G_{10}) &
 \end{aligned}$$

(3.8)

El resultado simplificado de $\mathcal{L}_2(FG)$, la ecuación (3.8), no puede simplificarse más. La mayoría de sus términos tienen factores residuales, es decir términos que no se prestan a una simplificación en forma de \mathcal{L}_k 's. Se discute este problema de la simplificación en los capítulos 7 y 9.

Capítulo 4. Cálculo de $\mathcal{L}_n(\mathcal{L}_m F)$ vía la Computación Directa.

4.a. Introducción.

El cálculo de $\mathcal{L}_n(\mathcal{L}_m F)$ es de una naturaleza simbólica. Se trata de la búsqueda de una representación polinomial en términos de los $\mathcal{L}_k F$'s, donde la función F es una función cualquiera infinitamente diferenciable. De la definición del operador \mathcal{L}_k se concluye que el resultado del cálculo de $\mathcal{L}_n(\mathcal{L}_m F)$ va a ser un conjunto de términos del tipo $F_{x(1),y(1)} F_{x(2),y(2)} \dots F_{x(i),y(i)}$, donde $i = (n+1) \cdot (m+1)$. Entonces hay que utilizar un lenguaje que permite el fácil manejo de estos términos. Otra característica que debe tener el lenguaje elegido es un código eficiente, ya que para n y m grandes se genera un número excesivo de términos y tarda mucho en analizar a todos. Se eligió el lenguaje de programación C debido a su facilidad para manejar estructuras de datos y a su eficiente código ejecutable. Un listado del programa se encuentra en el suplemento de la tesis.

El programa está separado en cuatro secciones, según los sucesivos pasos que hay que seguir para llegar hasta el resultado final. El primer paso calcula $\mathcal{L}_m F$ y guarda este conjunto de sub-términos que es del tipo $F_{x(0),y(0)} F_{x(1),y(1)} \dots F_{x(j),y(j)}$, $j = m+1$, en una estructura adecuada. Este paso es muy sencillo debido a que los términos vienen directamente de la definición de un determinante de orden m ,

$$\begin{vmatrix} F_{00} & F_{01} & \dots & F_{0m} \\ F_{10} & F_{11} & \dots & F_{1m} \\ \vdots & & \ddots & \\ F_{m0} & F_{m1} & & F_{mm} \end{vmatrix}$$

donde los términos son interpretados de la siguiente manera.

$$F_{0,y(0)} F_{1,y(1)} \dots F_{n,y(n)} = \partial x^0 \partial y^{y(0)} F \cdot \partial x^1 \partial y^{y(1)} F \dots \partial x^n \partial y^{y(n)} F$$

Surgen los demas pasos al iterar la operación \mathcal{L}_k , haciendo \mathcal{L}_n sobre $\mathcal{L}_m F$,

$$\begin{vmatrix} \langle \mathcal{L}_m F \rangle_{00} & \langle \mathcal{L}_m F \rangle_{01} & \dots & \langle \mathcal{L}_m F \rangle_{0n} \\ \langle \mathcal{L}_m F \rangle_{10} & \langle \mathcal{L}_m F \rangle_{11} & \dots & \langle \mathcal{L}_m F \rangle_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ \langle \mathcal{L}_m F \rangle_{n0} & \langle \mathcal{L}_m F \rangle_{n1} & \dots & \langle \mathcal{L}_m F \rangle_{nn} \end{vmatrix}$$

donde cada elemento del determinante se interpreta como $\partial x^{x(1)} \partial y^{y(j)} \langle \mathcal{L}_m F \rangle$. Se ve que los términos del resultado final vienen de construcciones como,

$$\langle \mathcal{L}_m F \rangle_{0,y(0)} \cdot \langle \mathcal{L}_m F \rangle_{1,y(1)} \dots \langle \mathcal{L}_m F \rangle_{n,y(n)} = \partial x^0 \partial y^{y(0)} \langle \mathcal{L}_m F \rangle \cdot \partial x^1 \partial y^{y(1)} \langle \mathcal{L}_m F \rangle \dots \partial x^n \partial y^{y(n)} \langle \mathcal{L}_m F \rangle$$

El paso 2 hace las derivadas parciales indicadas sobre los conjuntos de sub-términos representados por $\mathcal{L}_m F$. El paso 3 termina el cálculo de los términos del resultado final al hallar los productos entre los conjuntos $\partial x^{x(1)} \partial y^{y(j)} \langle \mathcal{L}_m F \rangle$. El paso 4 toma el resultado final y lo simplifica, reduciendo conjuntos de términos finales a $\mathcal{L}_k F$'s.

4.b. Codificación de $\mathcal{L}_k F$.

Para resolver determinantes partimos de su misma definición con la cual se puede hallar su valor,

$$\det A = \sum_{\pi} (\text{signo } \pi) a_{0\pi(0)} a_{1\pi(1)} \dots a_{n\pi(n)} \quad (4b.1)$$

donde $\pi(\text{elem})$ es una función uno-a-uno que mapea el conjunto de elementos $0, 1, \dots, n$ sobre sí mismo. Visto de otra manera, $\pi(\text{elem})$ produce permutaciones sobre el conjunto finito de elementos, $0, 1, \dots, n$.

Para encontrar los términos del determinante hay que sumar sobre todas las $(n + 1)!$ permutaciones posibles. El signo de cada término es positivo o negativo según si la cantidad de inversiones que habría que hacer sobre la permutación para recuperar el orden natural es de orden par o impar, respectivamente. Por inversión se entiende que hay uno o más elementos hacia la derecha que son de menor magnitud que el elemento bajo observación. Por ejemplo, $(0\ 3\ 1\ 2)$ tiene dos inversiones, ya que el 3 está fuera de orden con respecto al 1 y al 2.

Interpretando la ecuación 4a.1 como representación del operador \mathcal{L}_x resulta que,

$$\mathcal{L}_x F = \sum_{\pi} (\text{signo } \pi) \partial x^0 \partial y^{\pi(0)} F \cdot \partial x^1 \partial y^{\pi(1)} F \cdot \dots \cdot \partial x^k \partial y^{\pi(k)} F \quad (4b.2)$$

donde las potencias de derivadas parciales coinciden con las índices de las entradas al determinante.

La fórmula para \mathcal{L}_x en forma de determinante es muy fácil de traducir a una representación numérica para el cálculo de su valor por computadora. Lo único que hay que guardar son las permutaciones. En un arreglo con $K = k + 1$ entradas, el valor de la potencia de ∂x coincide con el índice del arreglo y el valor de la potencia de ∂y se guarda en dicho índice,

$$\text{arreglo}[\text{potencia}_{\partial x}] = \text{potencia}_{\partial y}$$

La estructura que almacena la representación de \mathcal{L}_x necesita memoria para $K!$ permutaciones. Y cada permutación requiere memoria para K valores enteros que representan a los K $\pi(i)$'s más un valor entero para el signo de la permutación,

```
char perms[K_Factorial][K + 1];
```

donde, para ahorrar espacio de memoria, se debe de usar un tipo de dato entero que sólo ocupa un byte (en este caso "char"), ya que los valores que se manejan son del orden de 10.

Hay muchas maneras de generar permutaciones. No se recomienda usar algoritmos recursivos, ya que el número de permutaciones, igual a $K!$, puede ser muy grande para K grande, y puede ocasionar sobreflujo a la pila durante la ejecución. Los métodos iterativos son mucho más seguros y son, generalmente, más rápidos también.

El algoritmo que se implementó empieza con los elementos en su orden natural: $0, 1, 2, \dots, k$ y con el signo positivo. La segunda permutación se consigue invirtiendo el elemento $k-1$ con el elemento k . Esta segunda permutación tiene signo negativo, ya que requiere 1 inversión (número impar) para recuperar el orden natural. Sobre estas dos primeras permutaciones se toma el elemento $k-2$ y lo recorre hacia la derecha cuantas veces se puede a través de sucesivas inversiones con los vecinos. Se repite con $k-3$ sobre todas las anteriores permutaciones, y etc., hasta que se haya recorrido el elemento 0. Observe el proceso con $k = 3$, que nos da $4! = 24$ permutaciones,

Orden Natural	2 →	1 →	0 →
+ 0 1 2 3	- 0 1 3 2	- 0 2 1 3	- 1 0 2 3
		+ 0 2 3 1	+ 1 2 0 3
		+ 0 3 1 2	- 1 2 3 0
		- 0 3 2 1	+ 1 0 3 2
			- 1 3 0 2
			+ 1 3 2 0
			+ 2 0 1 3
			- 2 1 0 3
			+ 2 1 3 0
			- 2 0 3 1
			+ 2 3 0 1
			- 2 3 1 0
			- 3 0 1 2
			+ 3 1 0 2
			- 3 1 2 0
			+ 3 0 2 1
			- 3 2 0 1
			+ 3 2 1 0

En el programa, la función `genera_perms()` toma un elemento del conjunto $0 \dots n$, `elem`, y una permutación de las generadas con un elemento anterior, `perm`, y los manda a la función `intcam()`. En `intcam()` se generan todas las posibles permutaciones nuevas al

recorrer el elemento elem hacia la derecha en la permutación perm.

La generación de los signos se hace en Intcam() de manera muy sencilla. Cada inversión sobre una permutación produce una nueva permutación con signo opuesto a la permutación original. Esto funciona porque cada inversión que se hace de un elemento con su vecino derecho es una inversión hacia mayor desorden, ya que estos vecinos siempre son de menor magnitud en el método empleado.

Para el cálculo de $\mathcal{L}_n(\mathcal{L}_m F)$ hacen falta dos juegos de permutaciones, uno sobre $N = n+1$ elementos y otro sobre $M = m+1$ elementos. Por lo tanto, en el programa se hizo una modificación a la estructura perms[[]], dejándolo como

```
char perms[2][NMmaxFactorial][NMmax + 1];
```

Las rutinas de generación de permutaciones también sufren ligeras modificaciones. En una sola llamada a genera_perms() se generan los M! y los N! permutaciones.

4.c. Derivación en \mathcal{L}_n aplicada sobre $\mathcal{L}_m F$.

$\mathcal{L}_m F$ quedó almacenado en perms[[]][[]]. El siguiente paso es aplicar las derivadas parciales representadas por \mathcal{L}_n sobre los términos de $\mathcal{L}_m F$. Tomando como ejemplo el caso de $n = 1$,

$$\mathcal{L}_1(\mathcal{L}_m F) = \begin{vmatrix} \mathcal{L}_m F & \partial_x \mathcal{L}_m F \\ \partial_y \mathcal{L}_m F & \partial_x \partial_y \mathcal{L}_m F \end{vmatrix} \quad (4c.1)$$

se ve que las derivaciones son sobre conjuntos de $(m + 1)!$ términos. Luego se ve que cada $\partial_x^i \partial_y^k \mathcal{L}_m F$ produce un factor que es, a su vez, un conjunto de términos. Ahora, los términos del resultado de \mathcal{L}_n son más complicados de hallar que en el caso de \mathcal{L}_m . No es simplemente cuestión de encadenar una hilera de F_{ik} 's. Es, más bien, cuestión de encadenar cadenas de F_{ik} 's. El trabajo de encadenar las cadenas se hace en el siguiente paso del programa. La tarea del paso bajo consideración es la de dejar las

sub-cadenas almacenadas para su uso posterior.

De la ecuación (4b.1) se ve que la primera permutación de \mathcal{L}_n da como "término" a $\{\mathcal{L}_n F\} * \{\partial x \partial y \mathcal{L}_n F\}$, donde $\{\}$ significa que es un conjunto de elementos. Siempre habrá $(n + 1)$ conjuntos de derivadas que tener que multiplicar entre sí. Hace falta hacer estructuras de datos que permitan almacenar estos conjuntos. En las variables `ders[0][0]` y `ll_der[0][0]` los índices de los factores se guardan en `f[2][0]` y el número de repeticiones del término se almacena en la variable "nveces".

```
struct d_ll {
    int nveces;
    char f[2][Mmax];
} ders[2][NDI], ll_der[NMAX][NDF]; /* NDI = # ders int's */
/* NDF = # ders finales */
```

En el programa se usa un espacio de memoria compartido para hacer las derivadas y otra para guardar a los $(n + 1)$ conjuntos $\{\}$. Esto se hace porque el $\{\}$ es de menor cantidad de términos que los conjuntos intermedios generados durante el proceso de las sucesivas iteraciones de las derivadas.

En `ll_der[0][0]` se guardan los $\{\}$ de un macro término. Luego, el paso posterior hace el producto de $\{\} * \{\} * \dots * \{\}$ y guarda los términos resultantes en otra estructura como parte del resultado final. Así queda listo `ll_der[0][0]` para aceptar los conjuntos, $\{\}$, del siguiente macro término.

En `ders[0][0]` se maneja el proceso matemático de la derivación. Primero se pone en `ders[0][0]` los términos generados por $\mathcal{L}_n F$ (transcritos directamente desde `perms[0][0][0]`.) Luego, se deriva con respecto a y , ∂y , las veces que indica su potencia y después se deriva con respecto a x , ∂x , las veces indicadas. Cada iteración produce un nuevo juego de términos. Por eso los términos resultantes de la primera iteración se guardan en `ders[1][0]`. Luego se deriva sobre los que están en `ders[1][0]` y el resultado se guarda en `ders[0][0]`. Así se van alternando los juegos de términos entre `ders[0][0]` y `ders[1][0]` hasta agotar las iteraciones de ∂y y de ∂x .

Después de cada iteración se simplifica el conjunto de

términos juntando los que son iguales. La cuenta de cuantas veces aparece cada término se mantiene en su respectiva variable `ders[][]`. n veces, lo cual es un miembro de la estructura de dato `d_ll`. Al terminar, se copia el conjunto de términos resultante a `ll_ders[i][j]`, donde `i` representa el índice del factor.

Factor $\partial x^m \partial y^n F$, donde $m = 1$, se desarrolla como sigue,

$$\partial x^1 \partial y^0 (F_{00} F_{11} - F_{01} F_{10}) = F_{10} F_{11} + F_{00} F_{21} - F_{01} F_{20} - F_{10} F_{11}$$

(4c.2)

lo cual se reduce a $F_{00} F_{21} - F_{01} F_{20}$. Los términos $+ F_{10} F_{11}$ y $- F_{10} F_{11}$ se eliminan de posteriores iteraciones, suponiendo que hubiera, al poner igual a 0 sus respectivos valores de número de repeticiones.

Para comparar un término con otro, los dos deben de estar ordenados. Por ejemplo, si se genera el término $F_{21} F_{00}$ hay que reordenarlo para que quede como $F_{00} F_{21}$. La convención que se está usando es de ordenar primero con respecto a `i` y luego con respecto a `k` en F_{ik} . Entonces el término $F_{12} F_{20}$ se queda como está, pero $F_{21} F_{02}$ se cambia a $F_{02} F_{21}$. La rutina `ord_ders()` es la que se encarga de este trabajo.

La rutina `borra_der_reps()` compara los términos (los sub-términos que resultan de las derivaciones) y elimina los repetidos a través de un manejo de la variable `n` veces de la estructura `d_ll`, tomando siempre en cuenta a los signos.

La operación de derivar parcialmente sobre una función es sumamente fácil cuando se considera una función en general. Por ejemplo, un factor F_{ik} , donde `i` es la potencia de la derivada parcial con respecto a `x`, y `k` es la potencia con respecto a `y`, se deriva haciendo sumas sobre `i` y `k`, $\partial x^3 \partial y^4 F_{11} = F_{45}$.

Siendo tan fácil la operación de derivación parcial, hay varias maneras para implementarlo en el programa. El método A itera las derivadas directamente sobre el conjunto $\{F\}$, ordenando términos y borrando repetidos después de cada iteración. El método B itera las derivadas indicadas sobre un solo término de la forma $F_{00} F_{00} \dots F_{00}$ de longitud $(m + 1)$ factores F_{00} , lo cual

produce un pequeño conjunto de términos. Después se suman todos estos términos resultantes a todos los términos del conjunto $\{Z_n F\}$ de la forma que se indica en las ecuaciones (4c.3) y (4c.4). En el método B el proceso de ordenar términos y de borrar repetidos se hace una sola vez al final de la operación de sumas, Π .

Iteración de derivadas sobre $F_{00} F_{00}$,

$$\partial x^2 \partial y^1 (F_{00} F_{00}) = 2F_{00} F_{21} + 4F_{10} F_{11} + 2F_{01} F_{20} \quad (4c.3)$$

La ecuación (4b.3) sirve para calcular directamente $\partial x^2 \partial y^1 (Z_1 F)$,

$$\begin{aligned} \partial x^2 \partial y^1 (Z_1 F) &= (F_{00} F_{11} - F_{01} F_{10}) \Pi \{2F_{00} F_{21} + 4F_{10} F_{11} + 2F_{01} F_{20}\} \\ &= (F_{00} F_{11} \Pi 2F_{00} F_{21}) + (F_{00} F_{11} \Pi 4F_{10} F_{11}) + (F_{00} F_{11} \Pi 2F_{01} F_{20}) \\ &+ (F_{01} F_{10} \Pi 2F_{00} F_{21}) + (F_{01} F_{10} \Pi 4F_{10} F_{11}) + (F_{01} F_{10} \Pi 2F_{01} F_{20}) \\ &= 2F_{00} F_{32} + 4F_{10} F_{22} + 2F_{01} F_{31} \\ &\quad - 2F_{01} F_{31} + 4F_{11} F_{21} + 2F_{02} F_{30} \end{aligned} \quad (4c.4)$$

donde Π es un operador que ahora definimos y que significa que se suman los índices correspondientes. Por ejemplo,

$$F_{00} F_{11} \Pi F_{01} F_{20} = F_{01} F_{31}$$

El método B aparentemente debería ser más eficiente ya que hace las derivadas iteradas sobre un solo término original $F_{00} \dots F_{00}$, y lo que sigue es solamente hacer sumas. Pero resulta que el conjunto de términos resultante de las derivaciones sobre $F_{00} \dots F_{00}$ es muy grande debido a una mínima reducción en número de términos durante las sucesivas iteraciones, lo cual implica un mal uso de tiempo y memoria.

Se concluye que el método A es mejor que el método B. Se muestra una comparación de los dos métodos en la tabla 4.1..

n	m	prods	Método A			Método B		
			NDI	NDF	tiempo	NDI	NDF	tiempo
1	1	8	4	2	-	8	2	-
1	2	72	18	6	-	54	6	< 1"
1	3	1152	96	24	6"	384	24	12.5"
1	4	28800	600	120	3.5'	>2000	-	-
2	1	192	8	8	2"	18	8	2"
2	2	5184	36	24	17"	216	24	23"
2	3	331776	192	96	23'18"	>2000	-	-
3	1	6144	16	8	29"	32	8	30"
3	2	1119744	108	54	1hr16'	-	-	-
4	1	552960	24	18	30'	-	-	-

Tabla 4.1

donde NDI significa el número de términos distinguibles máximo en los conjuntos de derivadas intermedias, NDF significa el número de términos distinguibles finales al terminar de hacer todas las iteraciones indicadas de las derivadas, el valor de "prods" es la cuenta de cuantos términos finales son construidos y los tiempos fueron medidos en una computadora personal (PC) de 10 Mhz.

Se ve en la tabla que el número de derivadas finales es la misma en los dos métodos. Donde hay gran diferencia es en el máximo número de derivadas intermedias que son generadas.

Para demostrar la magnitud de cada problema $\mathcal{L}_n(\mathcal{L}_m F)$, se muestra con "prods" la cantidad de términos que se mandan al resultado final. Se puede apreciar que el tiempo de ejecución depende más del número de productos que de la derivación. Esto se verá en más detalle en el siguiente paso del programa. Aquí es suficiente observar que el método modificado de derivación si hace más lento al programa. Cabe mencionar, también, que los tiempos expresados en la tabla no toman en cuenta el almacenamiento ni el manejo de los productos. Los tiempos reales que dan resultados simplificados a $\mathcal{L}_n(\mathcal{L}_m F)$ son mucho mayores.

Otra observación de la tabla 4.1 es que los valores de NDI y NDF representan al número de derivadas que no quedaron canceladas. Ya que las canceladas desperdician algo de memoria, es necesario pedir más memoria que lo indicado por estos valores en la Tabla

(4.1).

4.d. Productos.

Lo que se tiene hasta aquí son los factores $\{\partial x^j \partial y^k \pi^{(i)} \mathcal{L}_m F\}$ de una permutación dada guardadas en $ll_der[i][j]$. Falta hacer los productos entre estos factores donde $i = 0, \dots, n$. Los productos se almacenan en otra estructura de datos como parte del resultado final y $ll_der[i][j]$ queda libre para el cálculo de la siguiente permutación sobre los N elementos. Resumiendo,

$$\mathcal{L}_n(\mathcal{L}_m F) = \sum_{\pi} (\text{signo } \pi) \{\partial x^0 \partial y^{\pi(0)} \mathcal{L}_m F\} \cdot \dots \cdot \{\partial x^n \partial y^{\pi(n)} \mathcal{L}_m F\}$$

(4d.1)

donde $\{\}$ representa un conjunto de términos. Habrá un producto proveniente de cada posible combinación en donde se escoja un término de cada $\{\text{factor } i\}$. El algoritmo usado en la rutina `producto()` del programa enumera todas estas posibles combinaciones y hace un producto en base a cada uno. Visto en un ejemplo,

$$\begin{array}{cccccccccccc} 0 & 1 & 2 & 0 & 1 & 00 & 01 & 10 & 11 & 20 & 21 \\ \{a + b + c\} \cdot \{g + h\} & = & a \cdot g + a \cdot h + b \cdot g + b \cdot h + c \cdot g + c \cdot h \\ i = 0 & & i = 1 \end{array}$$

donde se construye la serie 00, 01, 10, 11, 20, 21 usando una base distinta para cada dígito del "número" y donde las letras representan a términos almacenados en $ll_der[i][j]$. El primer dígito (de la izquierda) se refiere a un elemento del conjunto $\{a,b,c\}$ y su base es 3. El segundo dígito se refiere a un elemento del conjunto $\{g,h\}$ y su base es 2.

La estructura que almacena el producto y los resultados es la siguiente,


```

struct prod {
    char f[2][Mmax * Nmax];
    int nveces;
} prod, result[NR]; /* NR es el # máx de resultados */

```

En el programa la rutina `producto()` escoge un índice de cada `{factor i}` y, en base a ellos, crea un producto usando la rutina `make_prod()`. La estructura `temp[Nmax]` es donde se guarda el "número" cuyos dígitos, `temp[i]`, se refieren a los términos en `ll_der[i][]`. La rutina `copiar()`, que pasa las derivadas a `ll_der[][]`, es la que se encarga de guardar los valores de las bases en la estructura `lbase[NMAX]`. Cada base `i` es simplemente el número de términos almacenados en `ll_der[i][]`.

Sólo existe un producto a la vez. Después de crearlo se hace el intento de incluirlo en una de las categorías que manejan las rutinas de simplificación. Si hay éxito, el producto se almacena en una de las estructuras del resultado simplificado, que se verá más adelante. Si no hay éxito, el producto se almacena en `result[]`. Luego, se procede a calcular el siguiente producto.

Como en el caso de las derivadas parciales, hay más de una manera de hallar los productos. Un método alternativo al que se usó hace el producto `{factor 0}`-`{factor 1}` y se guarda el conjunto resultante en `prods[0][]` de la estructura,

```

struct prod {
    char f[2][Mmax * Nmax];
    int nveces;
} prods[2][NP];

```

Allí se simplifica al quitar términos repetidos a través de un manejo de la variable "nveces" de la estructura "prod". Luego se multiplican los términos en `prods[0][]` por los de `{factor 2}` y los guarda en `prods[1][]`, y así alternando los valores entre `prods[0][]` y `prods[1][]` hasta agotar los `{factor i}`. Luego se manda todo el conjunto de productos resultantes a las rutinas de simplificación.

En el caso de $n = 3$, $m = 1$, este método alternativo tarda 10 minutos para llenar `result[]` con productos, sin intentos de

simplificar el resultado. Con el método original, usando enumeración de índices, se tarda sólo 29 segundos para lograr lo mismo.

La lentitud del método alternativo se debe a las muchas veces que se tiene que ordenar términos y buscar repeticiones. Se tiene que hacer estas dos cosas después de cada multiplicación por un {factor 1}. Y las búsquedas de repeticiones implican un recorrido de toda la lista de productos anteriores. En el método original se ordena una sola vez cada producto y cada producto provoca un sólo recorrido por la lista de anteriores productos para ver si es una repetición. En cuanto a ahorro de memoria, con el método original se usa una vez el espacio de la variable *prod*. En cambio, el método alternativo requiere muchas instancias de esta misma estructura.

4.e. Simplificación.

Con $n > 2$ los términos del resultado de $\mathcal{L}(\mathcal{L}F)$ suelen tener residuos, es decir, términos que no pueden ser incluidos en ningún \mathcal{L}_k . Por ejemplo consideremos el término

$$4 F_{01} F_{02} F_{10} F_{11} F_{11} F_{20} F_{22} F_{33} \quad (4e.1)$$

el cual puede ser escrito como

$$4 F_{01} F_{10} F_{22} F_{33} \cdot F_{02} F_{11} F_{20} \cdot F_{11} = 4 [\mathcal{L}_3 F] \cdot [\mathcal{L}_2 F] \cdot F_{11} \quad (4e.2)$$

donde $[\mathcal{L}_k F]$ significa que la cadena es una permutación que corresponde a $\mathcal{L}_k F$. En este sentido, F_{11} es un residuo y el término aparece 4 veces en el resultado.

Si hubiera $4! \cdot 3! = 144$ términos distinguibles que tuvieran la representación igual al lado derecho de la ecuación (4e.2), se podría reducir estos a la representación,

$$4 \mathcal{L}_3 F \cdot \mathcal{L}_2 F \cdot F_{11} \quad (4e.3)$$

La simplificación del resultado de $\mathcal{L}_n(\mathcal{L}_1 F)$, lo cual es un conjunto de términos como la ecuación (4e.1), consiste en dejarlo como un conjunto de fórmulas del tipo de la ecuación (4e.3). Se logra esto extrayendo cadenas correspondientes a $\mathcal{L}_k F$'s a los términos y agrupando los términos según la $\mathcal{L}_1 F \cdot \mathcal{L}_k F$ -residuo ó $\mathcal{L}_1 F$ -residuo a que corresponden. Se podría buscar grupos de cadenas de mayor orden, por ejemplo $\mathcal{L}_1 F \cdot \mathcal{L}_j F \cdot \mathcal{L}_k F$ -residuo, pero esto no fue necesario para el caso particular de $\mathcal{L}_3(\mathcal{L}_1 F)$ que se estudió.

Como está diseñado el programa, cada producto se manda a simplificación antes de proceder a calcular el siguiente producto. Esto hace que uno puede pedir que solo se almacenen términos que coinciden con una categoría de $\mathcal{L}_1 F \cdot \mathcal{L}_k F$ ó $\mathcal{L}_1 F$ dado en cada corrida del programa, eliminando así, en parte, los problemas de cantidad de memoria disponible. Se puede reducir aun más el "campo visual" del programa a pedir que sólo almacene los términos con un residuo dado. Así se hizo para calcular $\mathcal{L}_3(\mathcal{L}_1 F)$. Primero se guardó solamente residuos, agrupados según su categoría de $\mathcal{L}_1 F \cdot \mathcal{L}_k F$ ó $\mathcal{L}_1 F$. Luego, basándose en esta información, se fue pidiendo listas de términos completos según su categoría y su residuo.

Para resolver el problema de $\mathcal{L}_3(\mathcal{L}_1 F)$ primero se sacó la lista completa de residuos según categorías. A continuación se muestra una parte de dicha lista. El formato que usaremos en la presentación es el siguiente,

$$[\mathcal{L}_1 F][\mathcal{L}_k F] \text{ con residuos:} \\ \text{signo cuantos factor}_1 \text{ factor}_2 \dots$$

el cual debe interpretarse como sigue: después de extraer una permutación correspondiente a $\mathcal{L}_1 F$ y una correspondiente a $\mathcal{L}_k F$ de un término del resultado final quedarán los factores residuales que se listan. El signo del residuo viene del signo del término completo multiplicado por los signos de las dos permutaciones. El valor de "cuantos" cuenta el número de términos en el resultado final que contienen permutaciones $[\mathcal{L}_1 F][\mathcal{L}_k F]$ y que tienen un residuo con el signo indicado.

LISTA PARCIAL de RESIDUOS ($n = 3, m = 1$):

$$[\mathcal{L}_4 F][\mathcal{L}_0 F] \text{ con residuos:}$$

$$+ 120 \quad 00 \quad 00$$

$$[\mathcal{L}_3 F][\mathcal{L}_2 F] \text{ con residuos:}$$

$$+ 576 \quad 11$$

$$[\mathcal{L}_3 F][\mathcal{L}_1 F] \text{ con residuos:}$$

$$\begin{aligned} & - 212 \quad 01 \quad 32 \\ & - 212 \quad 10 \quad 23 \\ & + 212 \quad 00 \quad 33 \\ & + 116 \quad 13 \quad 20 \\ & - 116 \quad 03 \quad 30 \\ & + 116 \quad 02 \quad 31 \end{aligned}$$

$$[\mathcal{L}_3 F][\mathcal{L}_0 F] \text{ con residuos:}$$

$$+ 76 \quad 10 \quad 13 \quad 21$$

$$\begin{array}{c} : \\ : \\ : \end{array}$$

$$[\mathcal{L}_2 F][\mathcal{L}_2 F] \text{ con residuos:}$$

$$\begin{aligned} & + 72 \quad 01 \quad 43 \\ & + 72 \quad 10 \quad 34 \\ & + 36 \quad 00 \quad 44 \end{aligned}$$

$$\begin{array}{c} : \\ : \\ : \end{array}$$

En la lista parcial de residuos se ven algunas simplificaciones directamente. Están los 120 términos de $(\mathcal{L}_0 F)^3 \mathcal{L}_4 F$, por ejemplo, donde $(m+1)! = (4+1)! = 120$. Están los 576 = $4! \cdot 3! \cdot 4$ términos de $\mathcal{L}_3 F \mathcal{L}_2 F \cdot 11$, donde el número de repeticiones de cada término es 4. Hay 2 veces $\mathcal{L}_2 F \mathcal{L}_2 F \cdot (01 \ 43 + 10 \ 34)$ y una vez $\mathcal{L}_0 F \mathcal{L}_2 F \mathcal{L}_2 F \cdot 44$.

Si todos los resultados encajaran tan perfectamente sería fácil simplificar el resultado completo de $\mathcal{L}_n(\mathcal{L}_m F)$. Pero, ocurre un fenómeno que esconde algunas de las permutaciones $\mathcal{L}_m F$. Volviendo a la lista parcial de residuos, se nota que hay 76 veces $[\mathcal{L}_3 F][\mathcal{L}_0 F] \cdot (10 \ 13 \ 21)$. En el resultado final se calcula que debería de haber $96 = 4! \cdot 1 \cdot (4 \text{ veces})$ de estos términos. La pregunta es, ¿Donde están los otros $20 = 5 \cdot (4 \text{ veces})$ términos de $[\mathcal{L}_3 F][\mathcal{L}_0 F] \cdot (10 \ 13 \ 21)$?

Para empezar, resulta que ocho de sus supuestos términos pertenecen a $[\mathcal{L}_3 F] \cdot (03\ 10\ 10\ 21)$,

$$\begin{aligned}
 & 4 (00) \cdot (-03\ 10\ 21\ 32 \\
 & \quad + 03\ 10\ 22\ 31) \cdot (10\ 13\ 21) = \\
 & [\mathcal{L}_0 F] \quad [\mathcal{L}_3 F] \quad \text{residuo} \\
 \\
 & = -4 (+00\ 13\ 21\ 32 \\
 & \quad - 00\ 13\ 22\ 31) \cdot (03\ 10\ 10\ 21) \\
 & \quad [\mathcal{L}_3 F] \quad \text{residuo}
 \end{aligned}$$

Estos dos términos distinguibles aparecen con un número de repeticiones igual a 8, lo cual permite que entren a la cuenta de $4 \cdot \mathcal{L}_0 F \mathcal{L}_3 F \cdot (10\ 13\ 21)$ y de $-4 \cdot \mathcal{L}_3 F \cdot (03\ 10\ 10\ 21)$. Los 7 términos distinguibles, $7 \cdot 4 = 28$ términos total, que faltan para completar a $4 \cdot \mathcal{L}_0 F \mathcal{L}_3 F \cdot (10\ 13\ 21)$ se encuentran en $-4 \cdot [\mathcal{L}_1 F][\mathcal{L}_3 F] \cdot (13\ 20)$ y en $-4 \cdot [\mathcal{L}_1 F][\mathcal{L}_3 F] \cdot (10\ 23)$,

$$\begin{aligned}
 & 4 (00) \cdot (-02\ 11\ 20\ 33 \\
 & \quad + 03\ 11\ 20\ 32) \cdot (10\ 13\ 21) = \\
 & \quad \quad \quad -4 (+00\ 11) \cdot (+02\ 10\ 21\ 33 \\
 & \quad \quad \quad \quad - 03\ 10\ 21\ 33) \cdot (13\ 20)
 \end{aligned}$$

$$\begin{aligned}
 & 4 (00) \cdot (+01\ 12\ 20\ 33 \\
 & \quad - 01\ 13\ 20\ 32) \cdot (10\ 13\ 21) = \\
 & \quad \quad \quad -4 (-01\ 11) \cdot (-00\ 12\ 21\ 33 \\
 & \quad \quad \quad \quad + 00\ 13\ 21\ 32) \cdot (13\ 20)
 \end{aligned}$$

$$\begin{aligned}
 & 4 (00) \cdot (-00\ 11\ 23\ 32 \\
 & \quad + 02\ 11\ 23\ 30) \cdot (10\ 13\ 21) = \\
 & \quad \quad \quad -4 (+00\ 11) \cdot (+00\ 13\ 21\ 32 \\
 & \quad \quad \quad \quad - 02\ 13\ 21\ 30) \cdot (10\ 23)
 \end{aligned}$$

$$\begin{aligned}
 & 4 (00) \cdot (+01\ 10\ 23\ 32) \cdot (10\ 13\ 21) = \\
 & [\mathcal{L}_0 F] \quad [\mathcal{L}_3 F] \quad \text{residuo} \\
 \\
 & \quad \quad \quad -4 (-01\ 10) \cdot (+00\ 13\ 21\ 32) \cdot (10\ 23) \\
 & \quad \quad \quad [\mathcal{L}_1 F] \quad [\mathcal{L}_3 F] \quad \text{residuo}
 \end{aligned}$$

Otra cosa que ocurre es que algunos residuos se cancelan, aunque los términos completos no se cancelan. En el caso de $\mathcal{L}_3(\mathcal{L}_1 F)$ hubo dos términos distinguibles que se escondieron así,

$$\begin{array}{r} 4 (00) \cdot (+ 00 11) \cdot (+ 02 12 24 30 31) \\ 4 (00) \cdot (- 01 10) \cdot (- 02 12 24 30 31) \\ \mathcal{L}_0 F \quad \mathcal{L}_1 F \quad \text{residuo} \end{array}$$

En la lista de residuos para $[\mathcal{L}_0 F][\mathcal{L}_1 F]$ no aparece el residuo (02 12 24 30 31) debido a que hubo 4 positivos del mismo y 4 negativos, por lo cual se cancelaron.

Clarificando el asunto de los signos, el término tiene un signo que se almacena en su variable "nveces" lo cual guarda el número de repeticiones de cada término. Luego, el signo del residuo se calcula según lo que queda después de aplicar los signos correspondientes a las permutaciones sobre $\mathcal{L}_k F$ que se extraen del término. En el ejemplo de los residuos cancelados, si los residuos hubieran tenido el mismo signo se hubiera podido reducir los dos términos correspondientes a $\mathcal{L}_0 F \cdot \mathcal{L}_1 F \cdot (\text{resid})$.

El ejemplo de los residuos cancelados demuestra, además, que el resultado tiene algunos juegos incompletos de permutaciones. Debido a esto, no es posible simplificar el resultado sin ver a todos los términos en su forma completa para saber cuales faltan de los juegos de permutaciones.

Para la simplificación de $\mathcal{L}_3(\mathcal{L}_1 F)$, que se muestra en el capítulo 6, se estudió a cada uno de los 1200 términos distinguibles para asegurar que no hubiera error en la interpretación de las listas de residuos. Este labor costó unas 3 semanas de tiempo. No es factible pensar en usar este mismo método para ir avanzando al estudio de n's y m's de mayor magnitud. Faltaría elaborar más las rutinas de simplificación y muy probablemente usar más tiempo y memoria para extraer a todos los términos que corresponden a cada fórmula dada de $\mathcal{L}_{k(1)} \dots \mathcal{L}_{k(1)} \cdot \text{residuo}$.

El programa usa muchas rutinas para encontrar las cadenas y los residuos. No tiene caso hablar de todas ellas, pero si puede

ser de utilidad explicar el método que se usó.

Trabajemos sobre un ejemplo particular. Consideremos el siguiente término como ejemplo:

$$F_{01}^F F_{02}^F F_{10}^F F_{11}^F F_{11}^F F_{20}^F F_{22}^F F_{33}$$

el cual se representa por la lista

01 02 10 11 11 20 22 33

donde,

$$F_{x(i)y(i)} = \partial x^{x(i)} \partial y^{y(i)} F,$$

A las 8 parejas en la lista las denotaremos por las primeras letras del alfabeto: a b c d e f g h. Lo primero que hay que hacer es mapear los factores. El mapeo se guarda en las estructuras:

```
char quien[valor x(i)][#];
char valor[valor x(i)][#];
```

En quien[][] se guarda el nombre de cada factor y en valor[][] se guarda el valor y(i) de los correspondientes factores. Una mejor manera hubiera sido usando una estructura tal como,

```
struct facts {
    char nombre;
    char valor_y;
} factor[n+m][1];
```

ya que así sería más clara la relación entre quien[][] y valor[][].

En el programa se nombran los factores con números desde 1 hasta (n + 1) * (m + 1). Pero aquí, para mayor claridad, se describe con nombres alfabéticos desde 'a' hasta poner una letra a cada factor.

El mapeo del término ejemplo es, pues, el siguiente,

$$\begin{array}{r} \text{quien}[\][\] = \end{array} \begin{array}{c} 0 \ 1 \ 2 \ 3 \\ \hline 0 \ \left| \begin{array}{cccc} a & c & f & h \\ 1 & b & d & g \\ 2 & - & e & - \end{array} \right. \end{array} \quad \begin{array}{r} \text{valor}[\][\] = \end{array} \begin{array}{c} 0 \ 1 \ 2 \ 3 \\ \hline 0 \ \left| \begin{array}{cccc} 1 & 0 & 0 & 3 \\ 1 & 2 & 1 & 2 \\ 2 & - & 1 & - \end{array} \right. \end{array}$$

Para encontrar cadenas tipo $[Z_k F]$ basta con escoger un factor de cada una de las $k+1$ columnas del arreglo $\text{valor}[\][\]$ de tal manera que los valores cubran a una permutación de $0..k$. Por ejemplo, la cadena $a-c-g-h$ con valores correspondientes $1-0-2-3$, es una cadena $[Z_3 F]$. La rutina `make_chains()` del programa busca todas las cadenas que hay y los almacena en $\text{cadena}[\text{longitud } k][\text{cual}][i = 0..k]$.

Se usa el método de "back-tracking" para hallar las cadenas. Por ejemplo, después de encontrar a $b-d-f-h$ se regresa al nivel de f para ver si hay otra opción en vez de h . No la hay, entonces se regresa hasta d . Tampoco hay camino que no sea a través de f . Pero regresando desde e se encuentra que e es una alternativa de d . Progresando desde e se completa la cadena $b-e-f-h$.

Se calculan primero todas las cadenas de longitud $k = 0$, luego de $k = 1$, y etc., hasta calcular las de $k = n + m$.

Del término ejemplar se encuentran las siguientes cadenas,

$$\begin{aligned} k = 3: & \quad a-c-g-h, \quad b-d-f-h, \quad b-e-f-h \\ k = 2: & \quad a-c-g, \quad b-d-f, \quad b-e-f \\ k = 1: & \quad a-c, \quad b-c, \quad b-d, \quad b-e \end{aligned}$$

Después de tener calculadas las cadenas se hace necesario, para descomponer el término, buscar cadenas independientes (que no comparten factores). Luego, los factores que sobran forman el residuo. Haciendo este proceso sobre el ejemplo da:

$$\text{TERMINO EJEMPLO SIMPLIFICADO: } (a-c-g-h) \cdot (b-d-f) \cdot e \\ \quad \quad \quad [Z_3 F] \quad [Z_2 F] \quad F_{11}$$

El trabajo de las rutinas de simplificación es agrupar los términos según categorías de $[Z_k F]$'s y, a partir de allí, según sus residuos. Todo esto sólo sirve al usuario para tener más

control sobre el gran número de términos que se encuentran en el resultado. La reducción a grupos de \mathbb{Z} -F's se hace a mano por el usuario.

Capítulo 5. Cálculo de $\mathcal{L}_m(FG)$ vía la Computación Directa.

El cálculo de $\mathcal{L}_m(FG)$ es muy parecido al de $\mathcal{L}_n(\mathcal{L}_n F)$. Por lo tanto no hace falta entrar en tantos detalles en este capítulo. Solamente se va a mostrar los paralelismos entre las dos operaciones y las estructuras de datos que se usan en el programa.

En la representación de $\mathcal{L}_m(FG)$ como determinante,

$$\mathcal{L}_m(FG) = \begin{vmatrix} (FG)_{00} & (FG)_{01} & \dots & (FG)_{0m} \\ (FG)_{10} & (FG)_{11} & \dots & (FG)_{1m} \\ \vdots & & \ddots & \\ (FG)_{m0} & (FG)_{m1} & & (FG)_{mm} \end{vmatrix}$$

se puede apreciar que los factores $(FG)_{ik}$ de los términos del determinante ya son conjuntos de sub-términos,

$$\partial x^1 \partial y^0 (FG) = F_{10} G_{00} + F_{00} G_{10}$$

La operación de \mathcal{L}_m sobre FG es paralelo a la operación de \mathcal{L}_n sobre $\mathcal{L}_n F$. Ya que las funciones F y G son generales, el resultado no depende de que sean diferentes o iguales. Suponiendo el caso de que F y G sean la misma función, tanto $\mathcal{L}_m(FG)$ como $\mathcal{L}_n(\mathcal{L}_n F)$ se tratan de una operación \mathcal{L}_k sobre un conjunto de cadenas $F_{x(0),y(0)} F_{x(1),y(1)} \dots F_{x(k),y(k)}$. Con F distinto a G , la única diferencia es que la mitad de los factores serán derivadas parciales sobre G en vez de sobre F .

El programa sigue los mismos pasos que en el caso de $\mathcal{L}_n(\mathcal{L}_n F)$. El primer paso es generar $(m+1)!$ permutaciones para conocer los índices de los factores $(FG)_{ik}$. El segundo paso es derivar los factores. El tercer paso es hallar los productos entre los factores de cada término del determinante. No se hicieron rutinas de simplificación, ya que el número de términos en el resultado

final de interés, $m = 2$, fue pequeño y fácil de simplificar a mano.

Las estructuras usadas son muy similares a las que se usan para $\mathcal{L}_n(\mathcal{L}_m F)$. La única fuerte diferencia es que, en el caso de $\mathcal{L}_m(FG)$ hay que separar los factores $F_{x^{(1)},y^{(1)}}$ de los factores $G_{x^{(1)},y^{(1)}}$. Y la ordenación de los términos se hace ordenando los factores F por separado y los factores G por separado.

La estructura que almacena las derivadas sobre los factores $(FG)_{x,y}$ es,

```
struct d_fg {
    int nveces;
    char f[2];
    char g[2];
} fg_der[Mmax][N1], fg[2][N2];
```

El variable $fg[2][N2]$ se usa para analizar un solo factor. El resultado de la primera derivación se guarda en $fg[0][\]$. La segunda iteración pone sus resultados en $fg[1][\]$. Sobre los factores $F, G_{x,y, x,y}$ que están en $fg[1][\]$ se itera con respecto a x o a y , según lo indicado, y los resultados se guardan en $fg[0][\]$. Así se van alternando los resultados de las sucesivas iteraciones entre $fg[0][\]$ y $fg[1][\]$ hasta agotar las derivaciones indicadas. Una vez terminada la derivación sobre un factor i se guarda el conjunto de factores resultantes en $fg_der[i][\]$.

En $fg_der[\][\]$ se guarda el resultado de hacer todas las derivadas parciales indicadas sobre cada factor de una permutación. Sobre estos factores se hallan los productos antes de proceder a analizar la siguiente permutación.

La estructura que almacena los productos y el resultado final es,

```
struct prod {
    int nveces;
    char f[2][Mmax];
    char g[2][Mmax];
} prod, result[Rmax];
```

donde se ve que hay $M = m + 1$ factores $F, G_{x,y, x,y}$ en los términos del resultado final.

Un listado del programa se encuentra en el suplemento a esta tesis. Es relativamente corto y está bien documentado. Los nombres de las rutinas explican por sí mismas sus funciones.

Capítulo 6. Resultado Simplificado de $\mathcal{L}_3(\mathcal{L}_1 F)$.

El resultado que se da aquí consiste de exactamente 1200 términos distinguibles. Tomando en cuenta el número de repeticiones de cada término, el resultado tiene exactamente 3720 términos. Esta es la cantidad de términos que la computadora calcula en su resultado completo.

La simplificación solamente agrupa a los términos, donde se puede, en cadenas de operadores \mathcal{L}_k multiplicados por residuos. No hay ningún intento de completar artificialmente a alguna cadena de operadores, como lo que sería el caso de hacer,

$$4 \mathcal{L}_3 F \cdot (+ 00 11) * (- 03 30) = 4 \mathcal{L}_3 F \cdot \mathcal{L}_1 F * (- 03 30) + \\ 4 \mathcal{L}_3 F \cdot (- 01 10) * (+ 03 30)$$

donde "*" significa multiplicación por un eslabón $[\mathcal{L}_k F]$ ó un juego completo de permutaciones $\mathcal{L}_k F$, "##" significa multiplicación por uno ó más residuos, y $(\pm XY \dots XY)$ significa un sub_término $(\pm F_{xy} \dots F_{xy})$.

RESULTADO SIMPLIFICADO DE $\mathcal{L}_3(\mathcal{L}_1 F)$:

$$\begin{aligned} & 1 (L_0 F)^3 L_4 F \\ & 4 L_2 F L_3 F * (+ 11) \\ & 4 L_0 F L_1 F L_3 F * (+ 33) \\ & 4 L_1 F L_3 F * (- 01 32 - 10 23) \\ & 4 (L_0 F)^2 L_3 F * (- 13 31) \end{aligned}$$

$$\begin{aligned}
& 4 L_0 F L_3 F \cdot (+ 01 13 30 + 03 10 31 \\
& \quad + 10 13 21 + 01 12 31) \\
& 1 L_0 F L_2 F L_2 F \cdot (+ 44) \\
& 2 L_2 F L_2 F \cdot (+ 01 43 + 10 34) \\
& 1 L_0 F L_1 F L_2 F \cdot (- 24 42) \\
& 2 L_0 F L_1 F L_2 F \cdot (- 23 43 - 32 34) \\
& 2 L_1 F L_2 F \cdot (- 01 23 42 - 10 24 32) \\
& 1 (L_0 F)^2 L_2 F \cdot (- 14 22 41) \\
& 1 (L_0 F)^2 L_2 F \cdot (+ 12 24 41 + 14 21 42) \\
& 2 (L_0 F)^2 L_2 F \cdot (+ 12 31 34 + 13 21 43) \\
& 1 L_0 F L_2 F \cdot (+ 02 14 20 41) \\
& 1 L_0 F L_2 F \cdot (+ 01 14 22 40 + 04 10 22 41 \\
& \quad - 02 14 21 40 - 04 12 20 41 \\
& \quad - 02 10 24 41 - 01 14 20 42 \\
& \quad - 01 12 24 40 - 04 10 21 42) \\
& 2 L_0 F L_2 F \cdot (- 01 13 22 41 - 10 14 22 31 \\
& \quad + 10 12 24 31 + 01 13 21 42 \\
& \quad - 02 10 31 34 - 01 13 20 43 \\
& \quad + 10 14 21 32 + 01 12 23 41 \\
& \quad - 01 12 30 34 - 03 10 21 43) \\
& 2 L_0 F (L_1 F)^2 \cdot (+ 23 23 42 + 24 32 32) \\
& 2 (L_0 F)^2 L_1 F \cdot (- 12 23 23 41 - 14 21 32 32) \\
& 4 (L_0 F)^2 L_1 F \cdot (- 13 21 23 42 - 12 24 31 32) \\
& 2 L_0 F L_1 F \cdot (+ 02 10 23 23 41 + 01 14 20 32 32 \\
& \quad + 01 12 23 23 40 + 04 10 21 32 32)
\end{aligned}$$

$$\begin{aligned}
4 L_0^F L_1^F & \cdot (+ 03 10 21 23 42 + 01 12 24 20 32 \\
& + 01 13 20 23 42 + 02 10 24 31 32) \\
2 (L_0^F)^3 & \cdot (+ 12 12 24 31 31 + 13 13 21 21 42 \\
& - 12 14 22 31 31 - 13 13 21 22 41) \\
2 (L_0^F)^2 & \cdot (- 04 12 12 20 31 31 - 02 13 13 21 21 40 \\
& + 02 10 14 22 31 31 + 01 13 13 20 22 41 \\
& + 04 10 12 22 31 31 + 01 13 13 21 22 40 \\
& + 02 12 14 20 31 31 + 02 13 13 20 21 41) \\
4 (L_0^F)^2 & \cdot (- 02 10 12 24 31 31 - 01 13 13 20 21 42 \\
& + 01 12 14 22 30 31 + 03 10 13 21 22 41 \\
& - 01 12 12 24 30 31 - 03 10 13 21 21 42) \\
4 L_3^F & \cdot (- 03 10 10 21 - 01 01 12 30 \\
& + 03 10 11 20 + 01 02 11 30) \\
2 L_2^F & \cdot (- 04 10 11 22 30 - 01 03 11 22 40 \\
& + 04 10 12 21 30 + 01 03 12 21 40 \\
& - 04 10 10 21 32 - 01 01 12 23 40 \\
& + 04 10 11 20 32 + 01 02 11 23 40 \\
& + 01 02 13 20 41 + 02 10 14 20 31 \\
& + 02 10 11 24 30 + 01 03 11 20 42 \\
& - 02 10 10 24 31 - 01 01 13 20 42 \\
& + 04 10 10 22 31 + 01 01 13 22 40 \\
& - 04 10 12 20 31 - 01 02 13 21 40) \\
2 L_0^F & \cdot (- 01 01 13 13 20 22 40 - 02 04 10 10 22 31 31 \\
& + 01 01 13 13 20 20 42 + 02 02 10 10 24 31 31 \\
& - 01 01 12 14 22 30 30 - 03 03 10 10 21 22 41 \\
& + 01 01 12 12 24 30 30 + 03 03 10 10 21 21 32 \\
& - 01 02 13 13 20 20 41 - 02 02 10 14 20 31 31 \\
& + 01 02 13 13 20 21 40 + 02 04 10 12 20 31 31)
\end{aligned}$$

Los términos que siguen tienen juegos completos más juegos incompletos de permutaciones:

$$4 L_3^F (- 01 10) \cdot (+ 02 31 + 13 20)$$

$$4 L_3^F (+ 00 11) \cdot (- 03 30)$$

$$\begin{aligned}
& 2 L_2 F (-01 10 22 + 01 12 20) * (-03 41 - 14 30) \\
& 2 L_2 F (+00 11 22 - 00 12 21) * (-04 40) \\
& 1 L_2 F (+00 11) * (+04 20 42 + 02 24 40) \\
& 2 L_2 F (+00 11) * (+02 30 34 + 03 20 43) \\
& 2 L_2 F (-01 10) * (+12 24 30 + 03 21 42 \\
& \quad + 02 23 41 + 14 20 32) \\
& 2 L_1 F (+00 11) * (-02 23 23 40 - 04 20 32 32) \\
& 4 L_1 F (+00 11) * (-03 20 23 42 - 02 24 30 32) \\
& 2 (L_0 F)^2 [(+00 11 22 \\
& \quad - 01 10 22 \\
& \quad - 02 11 20) * (+14 31 32 + 13 23 41) \\
& \quad + (+02 10 21 \\
& \quad - 00 12 21 \\
& \quad + 01 12 20) * (-14 31 32 - 13 23 41)] \\
& 2 L_0 F [(-02 11 20 + 02 10 21 - 00 12 21) * (+04 10 31 32) + \\
& \quad (-01 10 22 + 01 12 20 + 00 11 22) * (-04 10 31 32)] \\
& 2 L_0 F [(-02 11 20 + 01 12 20 - 00 12 21) * (+01 13 23 40) + \\
& \quad (-01 10 22 + 02 10 21 + 00 11 22) * (-01 13 23 40)] \\
& 2 L_0 F [(+01 12 20 - 01 10 22 - 02 11 20) * (+04 12 30 31) + \\
& \quad (+00 11 22 - 00 12 21 + 02 10 21) * (-04 12 30 31)] \\
& 2 L_0 F [(+02 10 21 - 01 10 22 - 02 11 20) * (+03 13 21 40) + \\
& \quad (+00 11 22 - 00 12 21 + 01 12 20) * (-03 13 21 40)] \\
& 2 L_0 F [(+01 12 20 - 01 10 22 - 00 12 21) * (+03 13 20 41) + \\
& \quad (+00 11 22 + 02 10 21 - 02 11 20) * (-03 13 20 41)] \\
& 2 L_0 F [(+02 10 21 - 01 10 22 - 00 12 21) * (+02 14 30 31) + \\
& \quad (+00 11 22 + 01 12 20 - 02 11 20) * (-02 14 30 31)] \\
& 2 L_0 F [(+01 12 20 - 00 12 21 - 02 11 20) * (+01 14 30 32) + \\
& \quad (+00 11 22 - 01 10 22 + 02 10 21) * (-01 14 30 32)]
\end{aligned}$$

$$4 \begin{matrix} L \\ F \\ O \end{matrix} (+ 00 11) * (+ 03 13 20 21 42 + 02 12 24 30 31 \\ + 04 12 20 31 32 + 02 13 21 23 40 \\ + 02 14 21 30 32 + 03 12 20 23 41)$$

$$4 \begin{matrix} L \\ F \\ O \end{matrix} (- 01 10) * (- 03 13 20 21 42 - 02 12 24 30 31 \\ - 02 14 20 31 32 - 02 13 20 23 41)$$

Los términos que siguen contienen solamente juegos incompletos de permutaciones:

$$2 (+ 00 11) [(+ 01 12 20 \\ + 02 10 21 \\ - 02 11 20) * (- 03 23 40 - 04 30 32) \\ + (+ 00 11 22 \\ - 01 10 22 \\ - 00 12 21) * (+ 03 23 40 + 04 30 32)]$$

$$4 (- 01 10) (- 00 12 21) * (+ 03 23 40 + 04 30 32)$$

$$2 (+ 00 11 22 \\ - 00 12 21) * (- 02 04 11 30 30 - 03 03 11 20 40 \\ + 01 04 12 30 30 + 03 03 10 21 40)$$

$$2 (+ 00 11 22) * (+ 01 02 14 30 30 + 03 03 10 20 41)$$

$$2 (- 00 12 21) * (- 01 02 14 30 30 - 03 03 10 20 41)$$

$$4 (+ 00 11 22) * (+ 02 04 10 30 31 + 01 03 13 20 40)$$

$$2 (+ 00 11) * (- 02 02 14 21 30 30 - 03 03 12 20 20 41 \\ + 02 02 11 24 30 30 + 03 03 11 20 20 42)$$

$$4 (+ 00 11) * (- 01 02 12 24 30 30 - 03 03 10 20 21 42 \\ - 01 03 13 20 20 42 - 02 02 10 24 30 31)$$

El resultado calculado por vía directa de la computación de $\mathcal{L}_3(\mathcal{L}_1 F)$ coincide con el resultado parcial previsto en la regla LL.4 (ver capítulo 7, sección b),

$$\mathcal{L}_0 F \cdot \mathcal{L}_2(\mathcal{L}_1 F) \cdot F_{44} \quad (\text{de la regla LL.4}) =$$

$$\mathcal{L}_0 F \cdot (F^2 \mathcal{L}_3 F + (\mathcal{L}_2 F)^2) \cdot F_{44} \quad (\text{usando la ecuación (2.6)}) =$$

$$F^3 \mathcal{L}_3 F \cdot F_{44} + \mathcal{L}_0 F \cdot (\mathcal{L}_2 F)^2 \cdot F_{44}$$

donde $F^3 \mathcal{L}_3 F \cdot F_{44}$ queda contenido en $F^3 \mathcal{L}_4 F$.

Capítulo 7. Características de la Expansión de $\mathcal{L}_n(\mathcal{L}_m F)$.

7.a. Introducción.

El resultado de la operación $\mathcal{L}_n(\mathcal{L}_m F)$ es un conjunto de términos que comparten algunas cualidades. Siguiendo unas reglas de construcción se pueden hallar todos los posibles términos del resultado final. Se concluye que el problema de cómo construir el resultado final reside en saber cuál será el número de instancias de cada término que cumple con dichas reglas.

Otra característica de los términos del resultado de $\mathcal{L}_n(\mathcal{L}_m F)$ es que, generalmente, no coinciden con una representación polinomial. Como se vió en el capítulo 4, en la sección sobre simplificación, muchos términos tienen residuos que no pueden ser reducidos a cadenas de permutaciones tipo $[\mathcal{L}_k F]$. Habiendo términos con residuos en el resultado final, se hace imposible la representación polinomial del resultado. Se concluye que no existe una fórmula polinomial general para la expansión de $\mathcal{L}_n(\mathcal{L}_m F)$, ya que para $n > 2$ la mayoría de los términos en el resultado final contienen residuos. Sin embargo, si hay una representación parcial polinomial, lo cual se describe en el capítulo 8.

7.b. Reglas de Construcción.

Las reglas de construcción de términos vienen directamente de la definición del operador \mathcal{L}_m . Lo primero que se observa es que todos los términos de un resultado son de la misma longitud. Aplicando \mathcal{L}_m una vez sobre una función F da como resultado $(m + 1)!$ términos de longitud $m + 1$. Aplicando \mathcal{L}_m una segunda vez produce $(n + 1)!$ productos sobre $n + 1$ conjuntos de sub-términos:

de longitud $m + 1$, dando como resultado términos de longitud $(n + 1) \cdot (m + 1)$.

REGLA LL.1: La longitud de los términos es de $(n+1) \cdot (m+1)$ factores F_{xy} .

Otra cosa que se observa es que la suma sobre los índices es una constante para todos los términos de un resultado dado. Dichas sumas pueden ser calculados como se indica en la regla LL.2, donde los $N \cdot M$ factores ($N = n + 1$, $M = m + 1$) se representan como $F_{x(i)y(j)}$, $i = 1, \dots, N \cdot M$.

$$\begin{aligned} \text{REGLA LL.2: } \sum_{i=1}^{N \cdot M} x(i) &= \sum_{j=1}^{N \cdot M} y(j) = \sum_{i=0}^n (i + \sum_{j=0}^m j) \\ &= \frac{1}{2} N(n + m(m + 1)) \end{aligned}$$

El razonamiento para hallar la regla LL.2 es el siguiente. Aplicando \mathcal{L}_n una vez da términos donde las $x(i)$'s tienen valores de $x(i) = i$ (según que los factores se ordenan primero con respecto a las potencias de ∂x). Y los valores $y(j)$ dan la misma suma pero con el orden permutado sobre los factores. Al aplicar \mathcal{L}_n una segunda vez, habrá un incremento de 1 en la suma de $x(i)$'s para cada ∂x aplicado sobre un término, y un incremento de 1 en la suma de $y(j)$'s para cada ∂y . Y como las sumas de ∂x y de ∂y son constantes para una \mathcal{L}_n que se va a aplicar, entonces todos los términos recibirán el mismo aumento en sus índices $x(i)$ y $y(j)$.

Otra cosa que se puede sacar de la definición del operador \mathcal{L}_m es el rango de valores permitidos para los índices de los términos resultantes. Aplicando \mathcal{L}_m una vez sobre una función F se ve desde las permutaciones que el máximo valor de un sub-índice va a ser m . También se ve que el mínimo valor va a ser 0 y que todos los valores intermedios enteros son válidos. Ahora, aplicando una segunda operación \mathcal{L}_n sobre los términos del resultado de $\mathcal{L}_m F$, el máximo incremento en potencia de ∂x y de ∂y será n . Entonces el

máximo valor alcanzable de potencias de derivadas sobre F será $(m + n)$. Y, obviamente, el mínimo valor sigue siendo 0.

REGLA LL.3: El rango de índices $x(i)$ y $y(j)$ es $0, 1, \dots, n+m$.

La tabla 7.1 muestra algunos valores particulares calculados en base a las reglas 1, 2, y 3.

n	m	# de factores	$\sum x(i)$	max valor de índice
1	1	4	3	2
1	2	6	7	3
1	3	8	13	4
2	1	6	6	3
2	2	9	12	4
2	3	12	21	5
3	1	8	10	4
3	2	12	18	5
3	3	16	30	6
4	1	10	15	5
5	1	12	21	6

Tabla 7.1

El resultado general correspondiente a la ecuación (2.8) está basado también en la definición del operador \mathcal{L}_k . Presentado como regla, tenemos

$$\text{REGLA LL.4: } \mathcal{L}_n(\mathcal{L}_m F) = \mathcal{L}_{m-1} F \cdot \mathcal{L}_{n-1}(\mathcal{L}_m F) \cdot \partial_x^{(n+m)} \partial_y^{(n+m)} F + \dots$$

La regla 4 vuelve a referirse a términos que ya están cubiertos por las reglas 1, 2, y 3, pero da más información. Dice que todos los términos de esta fórmula aparecerán exactamente una vez.

Muchos de los términos que cumplen con las reglas 1, 2, y 3 son generados por la operación $\mathcal{L}_n(\mathcal{L}_m F)$ pero desaparecen del resultado final debido a que aparecen tantas veces con signo positivo como con signo negativo. Los que persisten en el

resultado final tienen una variedad de números de repeticiones. Para construir por completo el resultado final general a la operación $\mathcal{Z}_n(\mathcal{Z}_m F)$ se necesitan reglas para calcular el números de repeticiones para cada término que cumple con las reglas 1, 2, y 3. La tabla 6.2 da una idea de la magnitud de las cancelaciones que ocurren y demuestra que muchos de los términos que aparecen en el resultado final tienen números de repeticiones diferentes a ± 1 .

n	m	■ TOTAL DE TERMINOS GENERADOS	■ DE TERMINOS DISTINGUIBLES FINALES	■ TOTAL DE TERMINOS FINALES
1	1	8	6	6
2	1	192	45	60
3	1	6144	1200	3720
1	2	72	46	48
2	2	5184	348	480

Tabla 7.2

7.c. Representación Polinomial.

Además de las reglas de construcción, hay otra cosa muy importante que caracteriza a los términos del resultado de la operación $\mathcal{Z}_n(\mathcal{Z}_m F)$. Se trata de su representación como cadenas de $\{\mathcal{Z}_k F\}$'s, como se vió en la sección sobre simplificación del capítulo 4. Allí se vió que para $n = 3$ la mayoría de los términos contienen residuos. Es razonable suponer que, en general, para $n > 2$ la mayoría de los términos van a tener este comportamiento.

Recordando que el objetivo de la tesis es encontrar a una representación polinomial general para la iteración $\mathcal{Z}_n(\mathcal{Z}_m F)$, se ve que dicha representación polinomial no existe si es que no existe una manera de eliminar a los residuos que aparecen en el resultado.

Antes de demostrar que no existe una manera de eliminar a los residuos que persisten en el resultado final, es interesante hacer

notar que para $n < 3$ también hay residuos. Sólo que, en estos casos, los términos que los contienen se cancelan entre sí. Un ejemplo sencillo de un residuo que se cancela donde $n < 3$ es el caso de $\mathcal{L}_1(\mathcal{L}_1 F)$,

$$\mathcal{L}_1(\mathcal{L}_1 F) = \mathcal{L}_0 F \cdot \mathcal{L}_2 F + F_{01} F_{02} F_{10} F_{20} - F_{01} F_{02} F_{10} F_{20}$$

donde,

$$F_{01} F_{02} F_{10} F_{20} = (-F_{01} F_{10}) \cdot (-F_{02} F_{20}) = [\mathcal{L}_1 F] \cdot \text{residuo}$$

Hay dos posibles maneras de eliminar a los residuos que aparecen en un resultado final. Una es, simplemente, reagrupar a los factores de los términos de tal forma que ninguno queda fuera de un eslabón $[\mathcal{L}_k F]$. La otra manera es encontrar una forma "mágica" de combinar a los términos tal que se oculta la existencia de los residuos.

En la sección sobre simplificación del capítulo 4 se discutió el problema de los términos cuya forma de agruparse en sub-términos tipo $[\mathcal{L}_k F]$ no es única. Un ejemplo que se vió allí es el siguiente,

Ejemplo 7.1:

$$\begin{aligned} (+ 00) \cdot (+ 01 \ 10 \ 23 \ 32) \cdot (+ 10 \ 13 \ 21) &= \\ [\mathcal{L}_0 F] \quad [\mathcal{L}_3 F] \quad \text{residuo} & \\ (- 01 \ 10) \cdot (+ 00 \ 13 \ 21 \ 32) \cdot (- 10 \ 23) & \\ [\mathcal{L}_1 F] \quad [\mathcal{L}_3 F] \quad \text{residuo} & \end{aligned}$$

En el ejemplo 7.1, el residuo cambia de longitud 3 a longitud 2, pero sigue siendo un residuo. Una manera más convincente de verlo es agrupando los factores del ejemplo 7.1 según sus valores $x(i)$.

0	1	2	3
00	10	21	32
01	10	23	
	13		

donde se ve que sobra un factor $x(i) = 1$ para el cual no hay un compañero con $x(i) = 0$. Basta con este solo ejemplo para demostrar que existen términos que no tienen representación polinomial como cadenas de operadores $\{L_k F\}$ sin residuos.

Falta ver la posibilidad de agrupar los términos de tal manera que se esconden los residuos. Como el resultado es una suma de términos, se antoja hacer la agrupación en base a sumas y restas. El ejemplo 7.2, tomado del resultado simplificado de $L_2(L_1 F)$, ilustra la idea.

Ejemplo 7.2:

$$L_3 F \cdot (+ 00 11) \cdot (- 03 30) = L_3 F \cdot L_1 F \cdot (- 03 30) \\ L_3 F \cdot (- 01 10) \cdot (+ 03 30)$$

En el ejemplo 7.2 se pretende solamente desaparecer a los $\{L_k F\}$'s sueltos, y se ve que la suma y resta artificial no sirve de mucho. Se simplifica un término, pero aparece otro nuevo con el mismo problema.

La situación empeora si se trata de eliminar a un residuo no representable como un $\{L_k F\}$. Allí no hay ganancia. Se desaparece un término pero aparece otro igual. El ejemplo 6.3 lo demuestra, usando uno de los términos del ejemplo 7.2.

Ejemplo 7.3:

$$\begin{aligned}
 \mathcal{L}_3 F \cdot \mathcal{L}_1 F \cdot (-03\ 30) &= \mathcal{L}_3 F \cdot \mathcal{L}_1 F \cdot (-03\ 30) + \mathcal{L}_3 F \cdot \mathcal{L}_1 F \cdot (+03\ 30) \\
 &\quad - \mathcal{L}_3 F \cdot \mathcal{L}_1 F \cdot (+03\ 30) \\
 &= -\mathcal{L}_3 F \cdot \mathcal{L}_1 F \cdot (+03\ 30) \\
 &= \mathcal{L}_3 F \cdot \mathcal{L}_1 F \cdot (-03\ 30)
 \end{aligned}$$

7.d. Conclusión.

Se concluye que no hay manera de hacer desaparecer a los residuos que aparecen en el resultado final de $\mathcal{L}_3(\mathcal{L}_1 F)$. En consecuencia, es de suponer que tampoco hay manera de desaparecer a los residuos que aparecen cuando $n > 3$ y, entonces, que no existe una representación polinomial para los resultados de $\mathcal{L}_n(\mathcal{L}_m F)$ donde $n > 2$.

Se puede apreciar en la tabla 7.2 que el número de términos en el resultado final cuando $n > 2$ es muy grande. En el capítulo 6 se encuentra listado el resultado simplificado completo de $\mathcal{L}_3(\mathcal{L}_1 F)$. Se puede apreciar allí que hay muchos términos con $[\mathcal{L}_m F]$'s sueltos, y que la mayoría de los términos, en general, contienen residuos. Para poder construir el resultado completo en base a las reglas mencionadas sería necesario encontrar una regla que determina el número de apariciones de cada término y aplicarlo a todos los términos constructibles en base a las reglas 1, 2, y 3.

No parece ser fácil encontrar un conjunto de reglas, como la regla 4, con los cuales sea posible representar a todo el resultado simplificado general de $\mathcal{L}_n(\mathcal{L}_m F)$, ya que la mayoría de los términos tienen residuos complicados. Pero para los términos con residuos bien comportados, por ejemplo, donde los factores:

Para $x(i) = y(i)$ del residuo cumplen que $x(i) = y(i)$, existen algunas reglas que se encuentran a través de la simple observación de unos resultados simplificados dados. Un ejemplo de esto se da en el capítulo 8 donde se desarrolla un conjunto de reglas para el caso general de $m = 1$.

Capítulo 8. Simplificación Parcial a $\mathcal{L}_n(\mathcal{L}_1 F)$.

La simplificación general a $\mathcal{L}_n(\mathcal{L}_m F)$ es muy complicado debido a la gran cantidad de términos en el resultado para $n > 2$ y a que no todos los términos caen dentro de juegos completos de permutaciones sobre $\mathcal{L}_{k(0)} F \dots \mathcal{L}_{k(1)} F$ residuo. Sin embargo, se pueden dar algunas fórmulas con las cuales simplificar una parte del resultado general. Se presentan aquí unas fórmulas para el caso de $m = 1$ que fueron extraídas de los resultados donde $n = 1, 2, 3, 4, (5)$.

Algunos de los (R)esultados (P)arciales del resultado de la expansión de $\mathcal{L}_3(\mathcal{L}_1 F)$ resultan interesantes por su cercanía a las cadenas de \mathcal{L}_k 's puros, ya que sus residuos consisten solamente de factores F_{xy} donde $x = y$.

$$R.P. (n=3, m=1) 1: 1 (\mathcal{L}_0 F)^3 \mathcal{L}_4 F$$

$$R.P. (n=3, m=1) 2: 4 \mathcal{L}_2 F \mathcal{L}_3 F \cdot (+ 11)$$

$$R.P. (n=3, m=1) 3: 4 \mathcal{L}_0 F \mathcal{L}_1 F \mathcal{L}_3 F \cdot (+ 33)$$

$$R.P. (n=3, m=1) 4: 1 \mathcal{L}_0 F \mathcal{L}_2 F \mathcal{L}_2 F \cdot (+ 44)$$

Estos resultados hacen sospechar que son parte de alguna(s) fórmula(s) general(es). Basandose en los resultados para $n = 1, 2, 3$, se encontraron varias fórmulas generales para el caso de $m = 1$.

$$FORMULA 1: \mathcal{L}_n(\mathcal{L}_1 F) = 1 \cdot (\mathcal{L}_0 F)^n \cdot \mathcal{L}_{n+1} F + \dots$$

$$\text{FORMULA 2: } \mathcal{L}_n(\mathcal{L}_1 F) = (n-1)^2 \cdot (\mathcal{L}_0 F)^{n-2} \cdot \mathcal{L}_1 F \cdot \mathcal{L}_n F \cdot \partial x^n \partial y^n F + \dots$$

$$\text{FORMULA 3: } \mathcal{L}_n(\mathcal{L}_1 F) = (n-1)^{\rho(n)} \cdot (\mathcal{L}_0 F)^{n-3} \cdot \mathcal{L}_2 F \cdot \mathcal{L}_n F \cdot \partial x^{n-2} \partial y^{n-2} F \\ + \dots, \text{ donde } \rho(n) = \begin{cases} 0 & n \text{ impar} \\ 2 & n \text{ par} \end{cases}$$

La fórmula 1 viene directamente del resultado parcial R.P.1 y es una extensión de la regla LL.4. La fórmula 2 cubre al resultado parcial R.P.3 y la fórmula 3 cubre al resultado parcial R.P.2. No hay una fórmula nueva para el resultado parcial R.P.4, ya que queda cubierto por la regla LL.4.

Para verificar que las fórmulas 1, 2, y 3 cumplen para el caso de $n = 4$, $m = 1$, es necesario calcular una parte del resultado de $\mathcal{L}_4(\mathcal{L}_1 F)$ via la computación directa. Es suficiente para dicho proposito una lista de los residuos,

LISTA PARCIAL de RESIDUOS ($n = 4$, $m = 1$):

$$\begin{aligned} & [\mathcal{L}_5 F][\mathcal{L}_0 F] \text{ con residuos:} \\ & \qquad \qquad \qquad +720 \quad 00 \quad 00 \quad 00 \\ & [\mathcal{L}_4 F][\mathcal{L}_2 F] \text{ con residuos:} \\ & \qquad \qquad \qquad +1084 \quad 00 \quad 22 \\ & \qquad \qquad \qquad \vdots \\ & [\mathcal{L}_4 F][\mathcal{L}_1 F] \text{ con residuos:} \\ & \qquad \qquad \qquad +2268 \quad 00 \quad 00 \quad 44 \\ & \qquad \qquad \qquad \vdots \end{aligned}$$

donde, por ejemplo, 2268 se divide por $5! \cdot 2!$ para encontrar que hay 9 veces un juego completo de permutaciones sobre $[\mathcal{L}_4 F][\mathcal{L}_1 F]$. Los restantes 108 términos se supone que sirven para cadenas inferiores de $[\mathcal{L}_i F][\mathcal{L}_j F]$ ($j \leq i < 4$).

Se muestra una parte del resultado simplificado de $\mathcal{L}_4(\mathcal{L}_1 F)$ basado en la lista de residuos, poniendo énfasis en los términos cuyo residuo contiene solamente factores F_{xy} donde $x = y$,

RESULTADO SIMPLIFICADO PARCIAL de $\mathcal{L}_4(\mathcal{L}_1 F)$:

$$\begin{aligned}
 f & \\
 1 & (L_0 F)^4 L_5 F && \text{(FORMULA 1)} \\
 1 & L_0 F L_2 F L_4 F \cdot F_{22} && \text{(FORMULA 3)} \\
 9 & (L_0 F)^2 L_3 F L_4 F \cdot F_{44} && \text{(FORMULA 2)} \\
 12 & L_2 F L_4 F \cdot F_{11} F_{11} \\
 3 & L_0 F L_4 F \cdot F_{00} F_{00} F_{22} F_{33} \\
 4 & L_0 F L_4 F \cdot (- F_{00} F_{02} F_{20} F_{33} - F_{00} F_{03} F_{22} F_{30} \\
 & \quad + F_{00} F_{03} F_{20} F_{32} + F_{00} F_{02} F_{23} F_{30}) \\
 0 & L_0 F L_4 F \cdot F_{02} F_{03} F_{20} F_{30} \\
 12 & (L_3 F)^2 \cdot (F_{00} F_{33} + F_{11} F_{22} - F_{01} F_{32} - F_{10} F_{23}) \\
 18 & (L_3 F)^2 \cdot (F_{13} F_{20} - F_{03} F_{30} - F_{12} F_{21} + F_{02} F_{31})
 \end{aligned}$$

Con este resultado parcial se comprueba hasta $n = 4$ la validez de las fórmulas 1, 2, y 3.

Modificando la manera de representar a la fórmula 1 se puede ver que las 3 fórmulas pertenecen a una sola familia.

$$\begin{aligned}
 \text{REGLA LL.5: } \mathcal{L}_n(\mathcal{L}_1 F) &= (n-1)^0 \cdot (\mathcal{L}_0 F)^{n-1} \cdot \mathcal{L}_0 F \cdot \mathcal{L}_{n+1} F \\
 &+ (n-1)^2 \cdot (\mathcal{L}_0 F)^{n-2} \cdot \mathcal{L}_1 F \cdot \mathcal{L}_n F \cdot \partial x^n \partial y^n F \\
 &+ (n-1)^{\rho(n)} \cdot (\mathcal{L}_0 F)^{n-3} \cdot \mathcal{L}_2 F \cdot \mathcal{L}_n F \cdot \partial x^{n-2} \partial y^{n-2} F \\
 &+ \dots \quad (\text{Comprobado hasta } n=4)
 \end{aligned}$$

Es de sospechar que haya más miembros de esta familia para cuando n es mayor. Para comprobarlo se tendría que hallar una parte del resultado de $\mathcal{L}_5(\mathcal{L}_1 F)$, ya que $\dots \mathcal{L}_3 F \cdot \mathcal{L}_4 F \dots$ para $\mathcal{L}_4(\mathcal{L}_1 F)$ está en conflicto con la regla LL.2. Es muy probable que la fórmula siguiente sea miembro,

$$\text{FORMULA 4: } \mathcal{L}_n(\mathcal{L}_1 F) = \alpha(n) \cdot (\mathcal{L}_0 F)^{n-4} \cdot \mathcal{L}_3 F \cdot \mathcal{L}_n F \cdot \partial x^{n-5} \partial y^{n-5} F + \dots$$

$$\text{donde } n\text{veces} = \alpha(n) = \begin{cases} 0 & n=5 \\ 7 & n \geq 5 \end{cases}$$

La regla LL.5, con la posible inclusión de la fórmula 4, es la contribución ofrecida para la expansión general de $\mathcal{L}_n(\mathcal{L}_1 F)$.

Cabe dentro de lo posible que la regla 5 sea un caso particular de una regla general que sirve para $\mathcal{L}_n(\mathcal{L}_m F)$. Para verificar esto sería necesario analizar por lo menos los casos $n = 3$ y $n = 4$ con $m = 2$. En cualquier caso, dicha regla solo resuelve una pequeña parte del resultado final donde $n \geq 3$.

Capítulo 9. Características de la Expansión de $\mathcal{L}_m(FG)$.

9.a. Introducción.

El resultado de la operación $\mathcal{L}_m(FG)$ es, igual que $\mathcal{L}_m(\mathcal{L}_m F)$, un conjunto de términos que comparten algunas cualidades. Siguiendo unas reglas de construcción se pueden hallar todos los posibles términos del resultado final. Se concluye que el problema de cómo construir el resultado final reside en saber cuál será el número de instancias de cada término que cumple con dichas reglas.

Las reglas de construcción que se presentan en este capítulo vienen directamente de la definición de la operación $\mathcal{L}_m(FG)$. No se pretende dar aquí un algoritmo para conocer el número de repeticiones de cada término. Si es que existe uno ha de ser muy complicado, ya que en muchas ocasiones el mismo término aparece algunas veces con signo positivo y otras veces con signo negativo.

Igual que el resultado de $\mathcal{L}_m(\mathcal{L}_m F)$, el resultado de $\mathcal{L}_m(FG)$, en general, no coincide con una representación polinomial. Esto se debe a los factores residuales que persisten en los términos del resultado final cuando $m > 1$. En el capítulo 3 se puede observar que el resultado de $\mathcal{L}_2(FG)$ ya tiene el problema de los residuos. La imposibilidad de desaparecer a los factores residuales del resultado final de $\mathcal{L}_m(FG)$, $m > 1$, se demuestra usando los mismos argumentos que se dieron en el capítulo 7 sobre el resultado final general para $\mathcal{L}_m(\mathcal{L}_m F)$. Por lo tanto, no se van a repetir los argumentos en este capítulo.

9.b. Reglas de Construcción.

La longitud de los términos del resultado de $\mathcal{L}_m(FG)$ tiene que ser $(m + 1)$ factores $F G_{xy}$, debido a la definición de cómo construir los términos de un determinante. Una manera de verlo es,

$$F_{x(0),y(0)} G_{x(0),y(0)} \cdot \dots \cdot F_{x(m),y(m)} G_{x(m),y(m)}$$

donde los valores $x(i)$ de F no tienen que ser iguales a los de $x(i)$ de G , y lo mismo vale para los valores $y(i)$.

REGLA FG.1: La longitud de los términos es de $(m+1)$ factores $F_{xy} G_{xy}$.

Todos los términos del determinante $\mathcal{L}(FG)$ difieren tan sólo en que habrá permutaciones sobre los valores de los índices $y(i)$ de $(FG)_{x(i),y(i)}: (FG)_{0,0} (FG)_{1,1} \dots (FG)_{m,m}$. De allí se determinan dos cosas. La primera es que el máximo valor de la potencia de una derivada parcial, sea con respecto a x , $x(i)$, o con respecto a y , $y(i)$, es m .

REGLA FG.2: El rango de índices $x(i)$ y $y(j)$ es de $0, 1, \dots, m$.

La otra observación es acerca de las sumas sobre los índices. Resulta que la suma sobre todos los índices $x(i)$ es igual a la suma de los $y(i)$ y es un constante en función a m . Además, al hacer la suma no se distingue entre la función F y la función G .

$$\text{REGLA FG.3: } \sum_{i=1}^P x(i) = \sum_{j=1}^P y(j) = \sum_{i=0}^m i = 1/2 \cdot m(m+1) = 1/4 \cdot Pm,$$

$$P = 2(m+1).$$

9.c. Conclusión.

Con las tres reglas de construcción dadas en la sección 9.b se puede construir el conjunto de todos los posibles términos del resultado final a la operación $\mathcal{L}(FG)$. Si hubiera una manera de saber cuantas veces y con que signo aparecen los términos en cada instancia, se tendría un algoritmo bastante rápido para hallar el resultado final de $\mathcal{L}(FG)$.

Capítulo 10. Conclusiones.

En este trabajo se hicieron programas para manejar al operador \mathcal{L}_m simbólicamente, reduciéndolo a un conjunto de términos individuales. Analizando los términos, reglas para su construcción fueron halladas y los algoritmos de simplificación fueron usados para agrupar los términos en conjuntos de permutaciones correspondientes a $\mathcal{L}F$'s. En particular, para el caso de $\mathcal{L}_3(\mathcal{L}_1F)$, se encontró que la expansión simplificada consiste de 1200 términos distinguibles. De estos, la mayoría no se pueden representar como cadenas de permutaciones correspondientes a $\mathcal{L}F$'s debido a que contienen factores residuales.

En base a la ecuación (2.8), el resultado de $\mathcal{L}_3(\mathcal{L}_1F)$ puede escribirse como,

$$\mathcal{L}_3(\mathcal{L}_1F) = (\mathcal{L}_0F)^3\mathcal{L}_4F + \sum \text{términos con residuos}$$

Restando los 5! términos de $(\mathcal{L}_0F)^3\mathcal{L}_4F$ a 1200 se quedan 1080 términos distinguibles que no pueden ser expresados completamente como polinomios construidos con $\mathcal{L}F$'s, $m = 0, \dots, 4$.

Algunos de los residuos son bien comportados, como es el caso de $4\mathcal{L}_0F \cdot \mathcal{L}_1F \cdot \mathcal{L}_3F \cdot F_{33}$. Tales residuos pueden ser absorbidos por multiplicaciones por $\mathcal{L}F$'s adecuados, como se hizo para ir de la ecuación (2.8) a la ecuación (2.10). Pero otros residuos, tales como en $2\mathcal{L}_2F \cdot \mathcal{L}_2F \cdot F_{01} \cdot F_{43}$, no pueden ser absorbidos de esta manera.

Usando razonamientos similares, la expansión de $\mathcal{L}_2(FG)$ resultó ser,


$$\mathcal{L}_2(FG) = F^3\mathcal{L}_2G + G^3\mathcal{L}_2F + \sum \text{terminos con residuos}$$

donde la cuenta de términos distinguibles es 64.

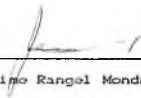
Expansiones no reducibles a una representación polinomial fueron hallados para casos particulares de $\mathcal{L}_3(\mathcal{L}_1 F)$ y para $\mathcal{L}_n(FG)$. Entonces se concluye que no existe una representación polinomial general para estas dos expresiones. La conjetura de que los "... términos de la ecuación (2.8) consisten de construcciones algebraicas de $\mathcal{L}_k F$, $m = 0, 1, \dots, k \leq 2(m+n) - 1$ se demostró que no es sostenible, debido a que multiplicaciones por $\mathcal{L}_k F$'s no pueden absorber a todos los residuos de los términos que los contienen en el caso particular $\mathcal{L}_3(\mathcal{L}_1 F)$. Sin embargo, si se encontraron unas fórmulas polinomiales y otras semi-polinomiales que corresponden a un gran número de los términos de la expansión general de $\mathcal{L}_n(\mathcal{L}_1 F)$.

Este trabajo demuestra solamente que una representación polinomial basada en multiplicaciones por $\mathcal{L}_k F$'s no existe para las expresiones $\mathcal{L}_n(\mathcal{L}_m F)$ y $\mathcal{L}_n(FG)$. Queda abierta la posibilidad de que exista otra representación no estrictamente polinomial basada en $\mathcal{L}_k F$'s con la cual se pueda representar a la expansión de estas dos expresiones generales.

El jurado designado por la Sección de Computación del Departamento de Ingeniería Eléctrica del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó esta tesis el 6 de septiembre de 1989.



Dr. Jerzy F. Plebański



Dr. Jaime Rangel Mondragón



Dr. Guillermo Morales Luna

