



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

7
2ej

FACULTAD DE INGENIERIA
DIVISION DE CIENCIAS DE LA TIERRA

MODELADO DEL POTENCIAL ELECTRICO EN
MEDIOS HORIZONTALES PARALELOS

T E S I S
QUE PARA OBTENER EL TITULO DE:
INGENIERO GEOFISICO
P R E S E N T A :
RICARDO ZAVALA TORRES

DIRECTOR DE TESIS: ING. FRANCISCO ARROYO CARRASCO

CD. UNIVERSITARIA,

1999

TESIS CON
FALLA DE ORIGEN

27/12/99



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL
AVENIDA DE
MEXICO

FACULTAD DE INGENIERIA
DIRECCION
60-I-004

SR. RICARDO ZAVALA TORRES
Presente

En atención a su solicitud, me es grato hacer de su conocimiento el tema que propuso el profesor Ing. Francisco Alejandro Arroyo Carrasco y que aprobó esta Dirección para que lo desarrolle usted como tesis de su examen profesional de Ingeniero Geofísico:

**MODELADO DEL POTENCIAL ELECTRICO EN MEDIOS HORIZONTALES
PARALELOS**

- I DIFERENCIAS FINITAS**
- II RED DE RESISTENCIAS ELECTRICAS**
- III APLICACIÓN DE LA TEORIA**
- IV CONCLUSIONES**
- V APENDICE**

Ruego a usted cumplir con la disposición de la Dirección General de la Administración Escolar en el sentido de que se imprima en lugar visible de cada ejemplar de la tesis el título de ésta.

Asimismo, le recuerdo que la Ley de Profesiones estipula que se deberá prestar servicio social durante un tiempo mínimo de seis meses como requisito para sustentar examen profesional.

Atentamente
"POR MI RAZA HABLARA EL ESPIRITU"
Cd. Universitaria, a 15 de enero de 1999
EL DIRECTOR

ING. GERARDO FERRANDO BRAVO

GFB*RLLR*gtg

**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**

**FACULTAD DE INGENIERÍA
DIVISIÓN DE CIENCIAS DE LA TIERRA**

Tesis para obtener el título de Ingeniero Geofísico.

**Modelado del potencial eléctrico en medios
horizontales paralelos**

Ricardo Zavala Torres

**PAGINACION
DISCONTINUA.**

AGRADECIMIENTOS

Quiero dar las gracias al Departamento de Geofísica de la División de Ciencias de la Tierra de la Facultad de Ingeniería de la UNAM por el apoyo brindado para la realización de este trabajo así como a familiares y amigos por toda la comprensión y paciencia que me mostraron, en especial agradezco a mi mamá que siempre esta apoyandome. También dedico este trabajo a la memoria de los que estuvieron conmigo pero ya no pudieron verme terminar.

Contenido

Lista de Figuras	3
1 Diferencias Finitas	5
1.1 Introducción	5
1.2 Diferencias Finitas: Caso Cilíndrico	7
1.3 Diferencias finitas en coordenadas esféricas	14
2 Red de Resistencias Eléctricas	27
2.1 Introducción	27
2.2 El Concepto de Resistencia	28
2.3 Teoría de circuitos	33
2.4 Uso de matrices para expresar las leyes de Kirchhoff y la relaciones de lazo.	33
2.5 La Teoría Matricial y Ley de Corrientes de Kirchhoff	40
2.6 Resistencias eléctricas con la malla cilíndrica.	41
2.7 Caso esférico	45
2.8 Análisis de error en una red eléctrica	49
2.9 Sensibilidad en una red eléctrica	52
2.10 La red adjunta como aproximación de sensibilidad	55
3 Aplicación de la Teoría	70
3.1 Introducción	70
3.2 Contrastes de resistividad	72
3.3 El Factor de Resistencia Terminal (FRT)	74
3.4 Cilíndrica vs esférica en la normal corta	76
4 Conclusiones	91
5 Apéndice	93
5.1 Medio con fronteras planas paralelas	93
5.1.1 Un plano de frontera.	93
5.1.2 Dos fronteras	98

Bibliografía

Lista de Figuras

1.1	Herramienta normal corta y malla cilíndrica	19
1.2	Malla cilíndrica en medios no coincidentes	20
1.3	Mallas cilíndrica y esférica	21
1.4	Sistema de Discretización	22
1.5	Malla de Diferencias Finitas Cilíndrica	23
1.6	Derivación en la frontera	24
1.7	Sistema Esférico	25
1.8	Malla Esférica	26
2.1	Tubo de Corriente	59
2.2	Lineas de corriente	60
2.3	Cilindro homogéneo	61
2.4	Sistema Cilíndrico	62
2.5	Circuito cilíndrico	63
2.6	Circuito esférico	64
2.7	Sistema esférico	65
2.8	Error en una celda	66
2.9	Circuito	67
2.10	Función error de R1	68
2.11	Función error de R3	69
3.1	Malla esférica en medio homogéneo	78
3.2	Disminución en el factor de expansión radial a 1.4	79
3.3	Disminución del factor radial a 1.15	80
3.4	Aumento en el factor de expansión radial	81
3.5	Radio interno de 0.1m	82
3.6	Incremento en el radio externo sin FRT	83
3.7	El factor de resistencia terminal	84
3.8	Malla cilíndrica	85
3.9	Error en la malla	86
3.10	Malla esférica	87

3.11	Contraste 2-500-2	88
3.12	Contraste 500-2-500	89
3.13	Contornos	90
5.1	Analítica frontera simple	104
5.2	Analítica dos fronteras	105

Capítulo 1

Diferencias Finitas

1.1 Introducción

En registros geofísicos la resistividad de la formación es un parámetro importante en la determinación de la saturación de hidrocarburos. La resistividad de la formación se mide al mandar corriente y medir la facilidad con la que fluye dentro de ella. El principio de medición de este sondeo eléctrico convencional se basa en introducir corrientes a la formación por medio de electrodos de corriente, y medir los voltajes entre los electrodos de medición, estos voltajes proporcionan la resistividad aparente para cada dispositivo, que se ve afectada por las resistividades y dimensiones geométricas de todos los medios al rededor del dispositivo (pozo, zona invadida y capas adyacentes).

En la herramienta normal corta se pasa una corriente de intensidad constante entre dos electrodos, A y B. La diferencia de potencial resultante se mide entre los dos electrodos, M y N. Los electrodos A y M se encuentran en la sonda. En teoría los electrodos B y N se localizan a una distancia infinita. En la práctica B es el blindaje del cable, y N es un electrodo en la brinda (el extremo inferior del cable que está cubierto de aislante) y están lejos de A y M. La distancia A y M se conoce como espaciamento (0.4064 m para la normal corta), el punto a donde se asigna la medición generalmente está en O, la mitad de la distancia entre A y M. En la presente tesis la medición se hace en el electrodo de potencial M, figura 1.1

La motivación de la tesis es la comprensión de los métodos de modelado y el efecto que tiene utilizar una malla geoméricamente no coincidente con el medio en 2D. Los métodos actuales de modelado se basan en una malla cilíndrica, coincidente con el problema de capas horizontales paralelas y pozo, pero en la realidad el pozo y capas no presentan esta idealizada geometría así como tampoco consideran problemas de fracturas o echado, es entonces la necesidad de resolver problemas de 3D o casos especiales de esta, lo que motiva este trabajo. La malla cilíndrica en un pozo fracturado y con capas inclinadas es una visión de lo que es una malla no coincidente con el medio figura 1.2.

Los objetivos son entonces: 1) Comprender y dominar el modelado clásico de la geometría cilíndrica para el medio coincidente con ella y 2) Desarrollar la malla esférica y confrontarla con el medio cilíndrico con el objeto de observar el efecto de una malla no coincidente con el medio, figura 1.3. La malla esférica se convierte entonces en un medio de investigación de los efectos de utilizar un geometría diferente a la del medio o desde el punto de vista de 3D, un medio diferente a la malla .

El problema es la determinación del potencial eléctrico en un medio con capas planas paralelas, así como comparar el efecto que tiene utilizar analogía con resistencias y diferencias finitas con las geometrías cilíndrica y esférica. El modelado se basan en la creación de una malla cilíndrica que resuelve el problema con diferencias finitas y con redes eléctricas. El problema directo implica desarrollar un modelo que proporcione la resistividad aparente del medio considerando el efecto del pozo, el cambio de resistividad, invasión, echado, etc. Es muy importante tener un modelo preciso que sea rápido para que una posterior inversión sea efectiva. En el método de resistencias eléctricas se utilizan las matrices de circuitos como instrumentos de análisis en un circuito, aunque su utilización puede ser no practica en la programación del algoritmo, a cambio ofrece la capacidad de obtener de una forma sencilla la sensibilidad de los elementos, corrientes y potenciales nodales así como la opción de ver la forma de la matriz de coeficientes con solo la matriz incidencia. El método de diferencias finitas ofrece otra forma de ver el problema y observar los efectos de analizar el mismo fenómeno con diferentes métodos y diferentes geometrías.

1.2 Diferencias Finitas: Caso Cilíndrico

Tradicionalmente el problema se resuelve partiendo de la ecuación de Maxwell

$$\nabla \times \mathbf{E} = 0 \quad (1.1)$$

que establece que el campo eléctrico \mathbf{E} generado por una fuente de corriente sin variación en el tiempo en un medio de conductividad σ es conservativo o irrotacional, por lo que puede ser derivado de un potencial escalar U

$$\mathbf{E} = -\nabla U \quad [V/m] \quad (1.2)$$

Además con la ley de Ohm para medios isótropos

$$\mathbf{J} = \sigma \mathbf{E} \quad [A/m^2] \quad (1.3)$$

que relaciona al vector densidad de corriente \mathbf{J} con el campo eléctrico.

Por otra parte la corriente generada por la fuente se define en terminos de el vector densidad de corriente y la normal de una superficie dada

$$I = - \oint_s \mathbf{J} \cdot d\mathbf{s} \quad [A]$$

que usando el teorema de la divergencia

$$- \oint_s \mathbf{J} \cdot d\mathbf{s} = \int_v (\nabla \cdot \mathbf{J}) dv$$

quedando

$$I = \int_v \nabla \cdot \mathbf{J} dv \quad [A]$$

por otro lado la corriente tambien se expresa en terminos de la variación de la carga por unidad de tiempo

$$I = \frac{dQ}{dt} = \frac{d}{dt} \int_v \rho_v dv = \int_v \frac{\partial \rho_v}{\partial t} dv \quad [A]$$

Las dos definiciones de corriente definen la relación entre la densidad de corriente y la variación de la carga contenida en dicho volumen, formando asi la ecuación de continuidad

$$\int_v (\nabla \cdot \mathbf{J}) dv = \int_v \frac{\partial \rho_v}{\partial t} dv \quad [A]$$

que también se cumple para un elemento diferencial de volumen Δv

$$(\nabla \cdot \mathbf{J}) \Delta v = \frac{\partial \rho_v}{\partial t} \Delta v \quad [A]$$

En todos los puntos de un medio de conductividad uniforme se cumple la ecuación de continuidad

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{J} = 0 \quad [A/m^3] \quad (1.4)$$

que se transforma en

$$\nabla \cdot \mathbf{J} = 0 \quad (1.5)$$

para todos los puntos del espacio excepto en la fuente.

Sustituyendo 1.3 y 1.2 en 1.5

$$\nabla \cdot (\sigma \nabla U) = 0 \quad (1.6)$$

que se cumple para todo el espacio excepto en la fuente.

La ecuación que relaciona el potencial y la fuente de corriente sin variación en el tiempo queda entonces

$$\nabla \cdot (\sigma \nabla U) + J = 0 \quad (1.7)$$

donde

σ es la conductividad medio [$\mathcal{U} \cdot m^{-1}$]

U es el potencial eléctrico [V]

J es la magnitud de la densidad de corriente eléctrica o la intensidad de la fuente de corriente por unidad de volumen, que existe solamente en el volumen en donde se encuentre concentrada la carga

$$J = \frac{\partial \rho}{\partial t} = \frac{I}{\Delta v} [A/m^3]$$

aclarando, que J utilizado de esta manera no es la densidad de corriente como se conoce, sino que es utilizada para efecto de discretización en diferencias finitas.

En coordenadas cilíndricas con simetría axial 1.7 queda :

$$\frac{1}{r} \frac{\partial}{\partial r} \left(\sigma r \frac{\partial U}{\partial r} \right) + \frac{\partial}{\partial z} \left(\sigma \frac{\partial U}{\partial z} \right) + J = 0 \quad (1.8)$$

donde

$$U = U(r, z) \text{ y } \sigma = \sigma(r, z)$$

El método de diferencias finitas es utilizado por Kim (1986) en coordenadas cilíndricas; para obtener el potencial eléctrico, en dicho trabajo se desarrolla la ecuación 1.8 con la expansión de la forma auto adjunta:

$$\frac{1}{r} \left(r \frac{\partial}{\partial r} \left(\sigma \frac{\partial U}{\partial r} \right) + \frac{\partial U}{\partial r} \frac{\partial(\sigma r)}{\partial r} \right) + \frac{\partial}{\partial z} \left(\sigma \frac{\partial U}{\partial z} \right) + J = 0$$

o

$$\frac{\partial}{\partial r} \left(\sigma \frac{\partial U}{\partial r} \right) + \frac{\sigma}{r} \frac{\partial U}{\partial r} + \frac{\partial}{\partial z} \left(\sigma \frac{\partial U}{\partial z} \right) + J = 0$$

Utilizando el sistema de discretización de la figura 1.4 que en diferencias finitas centrales queda

$$\begin{aligned} & \frac{\sigma_{i,j+1/2} \frac{U_{i,j+1} - U_{i,j}}{r_{j+1} - r_j} - \sigma_{i,j-1/2} \frac{U_{i,j} - U_{i,j-1}}{r_j - r_{j-1}}}{\frac{r_{j+1} - r_{j-1}}{2}} + \frac{\sigma_{i,j} U_{i,j+1} - U_{i,j-1}}{r_j r_{j+1} - r_{j-1}} + \\ & + \frac{\sigma_{i+1/2,j} \frac{U_{i+1,j} - U_{i,j}}{z_{i+1} - z_i} - \sigma_{i-1/2,j} \frac{U_{i,j} - U_{i-1,j}}{z_i - z_{i-1}}}{\frac{z_{i+1} - z_{i-1}}{2}} + J_{i,j} = 0 \end{aligned}$$

Esta forma ofrece buenos resultados, pero el punto principal de la tesis es explorar diferentes métodos para resolver el mismo problema, así como aprender la metodología de solución. Se utilizará la ecuación en diferencias finitas sin expansión. Esto significa que la derivación parcial en diferencias finitas se hace en la forma inicial de la ecuación de Laplace sin derivar el producto, obteniéndose así una simplificación de la misma, además de que en la ecuación para el sistema esférico que después se analizará, un desarrollo auto adjunto implica mayor número de términos trigonométricos, pero no existe diferencia en como se desarrollen las diferencias finitas

pues como se verá después la interpretación física de los coeficientes es lo que importa, entonces la forma sin expansión auto adjunta queda:

$$\frac{1}{r_j} \frac{r_{j+1/2} \sigma_{i,j+1/2} \frac{U_{i,j+1} - U_{i,j}}{r_{j+1} - r_j} - r_{j-1/2} \sigma_{i,j-1/2} \frac{U_{i,j} - U_{i,j-1}}{r_j - r_{j-1}}}{\frac{r_{j+1} - r_{j-1}}{2}} +$$

$$+ \frac{\sigma_{i+1/2,j} \frac{U_{i+1,j} - U_{i,j}}{z_{i+1} - z_i} - \sigma_{i-1/2,j} \frac{U_{i,j} - U_{i-1,j}}{z_i - z_{i-1}}}{\frac{z_{i+1} - z_{i-1}}{2}} + J_{i,j} = 0$$

o

$$\frac{1}{r_j} \left[\frac{2r_{j+1/2} \sigma_{i,j+1/2} (U_{i,j+1} - U_{i,j})}{(r_{j+1} - r_j)(r_{j+1} - r_{j-1})} + \frac{2r_{j-1/2} \sigma_{i,j-1/2} (U_{i,j-1} - U_{i,j})}{(r_j - r_{j-1})(r_{j+1} - r_{j-1})} \right] +$$

$$+ \frac{2\sigma_{i+1/2,j} (U_{i+1,j} - U_{i,j})}{(z_{i+1} - z_i)(z_{i+1} - z_{i-1})} + \frac{2\sigma_{i-1/2,j} (U_{i-1,j} - U_{i,j})}{(z_i - z_{i-1})(z_{i+1} - z_{i-1})} +$$

$$+ J_{i,j} = 0 \quad (1.9)$$

Una vez que se tienen las ecuaciones en diferencias finitas, es necesario agrupar los coeficientes y el término que multiplican para colocarlos en la matriz a resolver:

$$U_{i,j+1} \left[\frac{2r_{j+1/2} \sigma_{i,j+1/2}}{r_j (r_{j+1} - r_j) (r_{j+1} - r_{j-1})} \right], U_{i,j-1} \left[\frac{2r_{j-1/2} \sigma_{i,j+1/2}}{r_j (r_j - r_{j-1}) (r_{j+1} - r_{j-1})} \right]$$

$$U_{i+1,j} \left[\frac{2\sigma_{i+1/2,j}}{(z_{i+1} - z_i) (z_{i+1} - z_{i-1})} \right], U_{i-1,j} \left[\frac{2\sigma_{i-1/2,j}}{(z_i - z_{i-1}) (z_{i+1} - z_{i-1})} \right]$$

$$U_{i,j} \left[-\frac{2r_{j+1/2} \sigma_{i,j+1/2}}{r_j (r_{j+1} - r_j) (r_{j+1} - r_{j-1})} - \frac{2\sigma_{i+1/2,j}}{(z_{i+1} - z_i) (z_{i+1} - z_{i-1})} - \frac{2\sigma_{i-1/2,j}}{(z_i - z_{i-1}) (z_{i+1} - z_{i-1})} \right.$$

$$\left. - \frac{2r_{j-1/2} \sigma_{i,j+1/2}}{r_j (r_j - r_{j-1}) (r_{j+1} - r_{j-1})} \right]$$

En coordenadas cilíndricas la malla tendrá la forma de la figura 1.5 y condiciones de frontera siguientes:

La condición de frontera en el eje de simetría axial implica que los coeficientes $j+1$ y $j-1$ se modifican para cumplir con la restricción del no flujo de corriente excepto en la fuente.

Es necesario obtener los coeficientes para cada frontera pues son partes diferentes de la malla que no pueden ser desarrolladas con el caso general de 5 nodos, entonces para la frontera:

$$1 < i < i_{\max}, j = 0$$

La derivación en esta frontera solo afecta al término $i, j+1$, se puede proceder de dos maneras, haciendo un punto imagen del punto $(i, j+1)$ del otro lado de la frontera como en la figura 1.6 y desarrollar por diferencias centrales la segunda derivada en (i, j) o con la derivada segunda en el punto $i, j+1/4$, esta última ilustra la incorporación de la condición de frontera $D[u, r]=0$, mientras que la primera da por hecho esta condición; ambas formas dan el mismo resultado, para la segunda forma es:

$$\frac{1}{r} \frac{\partial}{\partial r} \left(r \sigma \frac{\partial U}{\partial r} \right)_{i,0} \simeq \frac{1}{r_0} \frac{\left(r \sigma \frac{\partial U}{\partial r} \right)_{i,1/2} - \left(r \sigma \frac{\partial U}{\partial r} \right)_{i,0}}{(r_1 - r_0)/2}$$

pero

$$\frac{\partial U}{\partial r}_{i,0} = 0$$

quedando

$$\frac{1}{r} \frac{\partial}{\partial r} \left(r \sigma \frac{\partial U}{\partial r} \right)_{i,0} \simeq \frac{1}{r_0} \frac{r_{1/2} \sigma_{i,1/2} (U_{i,1} - U_{i,0})}{(r_1 - r_0)/2}$$

Agrupando terminos

$$U_{i,1} \left[\frac{2r_{1/2}\sigma_{i,1/2}}{r_0(r_1 - r_0)^2} \right]$$

$$U_{i+1,0} \left[\frac{2\sigma_{i+1/2,0}}{(z_{i+1} - z_i)(z_{i+1} - z_{i-1})} \right], U_{i-1,0} \left[\frac{2\sigma_{i-1/2,0}}{(z_i - z_{i-1})(z_{i+1} - z_{i-1})} \right]$$

$$U_{i,0} \left[-\frac{2r_{1/2}\sigma_{i,1/2}}{r_0(r_1 - r_0)^2} - \frac{2\sigma_{i+1/2,0}}{(z_{i+1} - z_i)(z_{i+1} - z_{i-1})} - \frac{2\sigma_{i-1/2,0}}{(z_i - z_{i-1})(z_{i+1} - z_{i-1})} \right]$$

Que se agrupan en la matriz utilizando los índices $i=0,\dots,n, j=0,\dots,n$.

$$\begin{array}{l}
 \text{Ecuación de el nodo 1} \\
 \cdot \\
 \cdot \\
 \cdot \\
 \text{Ecuación de el nodo } n
 \end{array}
 \begin{array}{l}
 \text{nodo} \\
 0,0 \\
 1,0 \\
 2,0 \\
 \cdot \\
 n
 \end{array}
 \left[\begin{array}{ccccc}
 0,0 & 0,1 & 0 & 0,3 & 0 \\
 1,0 & \cdot & \cdot & 0 & i,j+1 \\
 0 & \cdot & \cdot & \cdot & 0 \\
 \cdot & 0 & \cdot & \cdot & i+1,j \\
 0 & i,j+1 & 0 & i-1,j & i,j
 \end{array} \right]$$

Entonces se tiene el sistema de ecuaciones:

$$\mathbf{A} \cdot \mathbf{u} = \mathbf{b}$$

La matriz generada es no simétrica y aunque existen metodos de solución para esta clase de problemas, al resolver se tienen valores negativos en las fronteras, entonces se utiliza la propiedad de hacer simétrica la matriz de coeficientes A , al multiplicar por el volumen de celdas:

$$V(i,j) = 2\pi r_j \left(\frac{r_{j+1} - r_{j-1}}{2} \right) \left(\frac{z_{i+1} - z_{i-1}}{2} \right)$$

Que reorganizando los extremos nodales y el centro para el caso general queda

$$(U_{i,j+1} - U_{i,j}) \left[\sigma_{i,j+1/2} \pi (z_{i+1} - z_{i-1}) \frac{r_{j+1}}{2(r_{j+1} - r_j)} \right] \quad [U]$$

$$(U_{i,j-1} - U_{i,j}) \left[\sigma_{i,j-1/2} \pi (z_{i+1} - z_{i-1}) \frac{r_{j-1}}{2(r_j - r_{j-1})} \right] \quad [U]$$

$$(U_{i+1,j} - U_{i,j}) \left[\frac{\sigma_{i+1/2,j} \pi r_j (r_{j+1} - r_{j-1})}{(z_{i+1} - z_i)} \right] \quad [U]$$

$$(U_{i-1,j} - U_{i,j}) \left[\frac{\sigma_{i-1/2,j} \pi r_j (r_{j+1} - r_{j-1})}{(z_i - z_{i-1})} \right] \quad [U]$$

y para la frontera

$$(U_{i,1} - U_{i,0}) \left[\left(\frac{r_1 + r_0}{r_1 - r_0} \right) \pi \left(\frac{z_{i+1} - z_{i-1}}{2} \right) \sigma_{i,1/2} \right] \quad [U]$$

$$I_{i,j} = J_{i,j} V_{i,j} = \frac{I}{V_{i,j}} V_{i,j} = I \quad [A]$$

En la ecuación de diferencias finitas puede ser agrupada:

$$\frac{(U_{i,j+1} - U_{i,j})}{R_a} + \frac{(U_{i,j-1} - U_{i,j})}{R_b} + \frac{(U_{i+1,j} - U_{i,j})}{R_c} + \frac{(U_{i-1,j} - U_{i,j})}{R_d} + I_{i,j} = 0$$

Observando las dimensiones de los coeficientes en su forma final puede verse su analogía con resistencias eléctricas. Esta última ecuación muestra cómo las diferencias finitas (DF) y la ley de corrientes de Kirchhoff que establece que la suma de corrientes que entran a un nodo son cero pueden resolver el mismo problema. Las ecuaciones en DF formarán la matriz pentadiagonal simétrica positiva que es fácilmente resuelta por el método de Choleski.

El lado izquierdo del sistema solo contiene al elemento al que esta aplicada la corriente, pero este nodo debido a que depende de la densidad de corriente de volumen

$$J = \frac{I \text{ amperes}[A]}{V \text{ volumen}[m^3]}$$

tiene que ser tratado con cuidado pues de este nodo depende la escala a la que esta la corriente, aunque no afecta la forma de la salida, sí su correcto valor.

En forma matricial el proceso queda:

Diferencias finitas puras

$$\begin{bmatrix} -a_{i,j} & c_{i+1,j} & 0 & b_{i,j+1} & 0 \\ d_{i-1,j} & -a_{i,j} & c_{i+1,j} & 0 & b_{i,j+1} \\ 0 & d_{i-1,j} & -a_{i,j} & c_{i+1,j} & 0 \\ e_{i,j-1} & 0 & d_{i-1,j} & -a_{i,j} & c_{i+1,j} \\ 0 & e_{i,j-1} & 0 & d_{i-1,j} & -a_{i,j} \end{bmatrix} \cdot \begin{bmatrix} u_{i,j+1} \\ u_{i+1,j} \\ u_{i,j} \\ u_{i-1,j} \\ u_{i,j-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ J_{i,j} \\ 0 \\ 0 \end{bmatrix}$$

al multiplicar por el volumen de cada celda a ambos lados de la ecuación, se tiene

$$(Diag(V) \cdot A) \cdot u = Diag(V) \cdot b$$

donde

$$\text{Diag}(V) \cdot b = J_{i,j} V_{i,j} = \frac{I}{V_{i,j}} V_{i,j} = I \text{ [A]}$$

que establece que en el lado izquierdo de la ecuación solamente va la intensidad de corriente y no es multiplicada por ningun otro factor o volumen.

$$\begin{bmatrix} v_{i,j} a_{i,j} & -v_{i+1,j} c_{i+1,j} & 0 & -v_{i,j+1} b_{i,j+1} & 0 \\ -v_{i+1,j} c_{i+1,j} & v_{i,j} a_{i,j} & -v_{i+1,j} c_{i+1,j} & 0 & -v_{i,j+1} b_{i,j+1} \\ 0 & -v_{i+1,j} c_{i+1,j} & v_{i,j} a_{i,j} & -v_{i+1,j} c_{i+1,j} & 0 \\ -v_{i,j+1} b_{i,j+1} & 0 & -v_{i+1,j} c_{i+1,j} & v_{i,j} a_{i,j} & -v_{i+1,j} c_{i+1,j} \\ 0 & -v_{i,j+1} b_{i,j+1} & 0 & -v_{i+1,j} c_{i+1,j} & v_{i,j} a_{i,j} \end{bmatrix} \cdot \begin{bmatrix} u_{i,j+1} \\ u_{i+1,j} \\ u_{i,j} \\ u_{i-1,j} \\ u_{i,j-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ I_{i,j} \\ 0 \\ 0 \end{bmatrix}$$

Que tiene la propiedad de ser simétrica positiva y puede ser resuelta con el algoritmo de Choleski

1.3 Diferencias finitas en coordenadas esféricas

La ecuación 5.2 en coordenadas esféricas es:

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left(\sigma r^2 \frac{\partial U}{\partial r} \right) + \frac{1}{r^2 \sin \phi} \frac{\partial}{\partial \phi} \left(\sigma \sin \phi \frac{\partial U}{\partial \phi} \right) + J = 0$$

Con el sistema de la figura 1.7

Que en diferencias finitas centrales y sin expansión auto-adjunta queda:

$$U = U(r, \phi) \text{ y } \sigma = \sigma(r, \phi)$$

$$\begin{aligned} & \frac{1}{r_j^2} \frac{\sigma_{i,j+1/2} r_{j+1/2}^2 \frac{U_{i,j+1} - U_{i,j}}{r_{j+1} - r_j} - \sigma_{i,j-1/2} r_{j-1/2}^2 \frac{U_{i,j} - U_{i,j-1}}{r_j - r_{j-1}}}{\frac{r_{j+1} - r_{j-1}}{2}} + \\ & + \frac{1}{r_j^2 \sin \phi_i} \frac{\sigma_{i+1/2,j} \sin \phi_{i+1/2} \frac{U_{i+1,j} - U_{i,j}}{\phi_{i+1} - \phi_i} - \sigma_{i-1/2,j} \sin \phi_{i-1/2} \frac{U_{i,j} - U_{i-1,j}}{\phi_i - \phi_{i-1}}}{\frac{\phi_{i+1} - \phi_{i-1}}{2}} + \\ & + J_{i,j} = 0 \end{aligned}$$

Las condiciones de frontera para el caso esférico, varían del caso cilíndrico en la condición de no existencia de corriente en las direcciones azimutales, implicando

que el algoritmo de formación de la matriz de coeficientes a resolver, no puede ser utilizado por las geometrías cilíndrica y esférica.

La malla esférica puede verse en la figura 1.8

Para el caso esférico siendo $\phi = 180^\circ$ y $\phi = 0^\circ$ el eje del pozo, las condiciones de frontera son:

a) En el eje de simetría axial $\phi = 0^\circ$ y $\phi = 180^\circ$

$$\frac{\partial U}{\partial \phi} = 0$$

que es la condición de frontera para la no existencia de corrientes azimutales.

b) Considerando un radio interno R_1 y el externo R_2 se tiene

En R_1

$$\frac{\partial U}{\partial r} = 0, \text{ excepto en donde esta localizada la fuente}$$

En R_2

$$U_{R_2} = 0$$

Los coeficientes esféricos $U_{i+1,j}, U_{i-1,j}, U_{i,j}, U_{i,j+1}, U_{i,j-1}$ quedan:

$$U_{i+1,j} \left[\frac{2 \sin \phi_{i+1/2} \sigma_{i+1/2,j}}{(\phi_{i+1} - \phi_i) (\phi_{i+1} - \phi_{i-1}) r_j^2 \sin \phi_i} \right]$$

$$U_{i-1,j} \left[\frac{2 \sin \phi_{i-1/2} \sigma_{i-1/2,j}}{(\phi_i - \phi_{i-1}) (\phi_{i+1} - \phi_{i-1}) r_j^2 \sin \phi_i} \right]$$

$$U_{i,j+1} \left[\frac{2r_{j+1/2}^2 \sigma_{i,j+1/2}}{r_j^2 (r_{j+1} - r_{j-1}) (r_{j+1} - r_j)} \right]$$

$$U_{i,j-1} \left[\frac{2r_{j-1/2}^2 \sigma_{i,j-1/2}}{r_j^2 (r_{j+1} - r_{j-1}) (r_j - r_{j-1})} \right]$$

$$U_{i,j} \left[-\frac{2}{r_j^2 \sin \phi_i (\phi_{i+1} - \phi_{i-1})} \left(\frac{\sin \phi_{i+1/2} \sigma_{i+1/2,j}}{(\phi_{i+1} - \phi_i)} + \frac{\sin \phi_{i-1/2} \sigma_{i-1/2,j}}{(\phi_i - \phi_{i-1})} \right) \right. \\ \left. - \frac{2}{r_j^2 (r_{j+1} - r_{j-1})} \left(\frac{r_{j+1/2}^2 \sigma_{i,j+1/2}}{(r_{j+1} - r_j)} + \frac{r_{j-1/2}^2 \sigma_{i,j-1/2}}{(r_j - r_{j-1})} \right) \right]$$

al multiplicar por el volumen :

$$V(i, j) = 2\pi r_j^2 \sin \phi_i \left(\frac{r_{j+1} - r_{j-1}}{2} \right) \left(\frac{\phi_{i+1} - \phi_{i-1}}{2} \right)$$

Agrupando los coeficientes de los nodos externos y el nodo central

$$(U_{i+1,j} - U_{i,j}) \left[\frac{\pi \sin(\phi_{i+1} - \phi_i) (r_{j+1} - r_{j-1}) \sigma_{i+1/2,j}}{2(\phi_{i+1} - \phi_i)} \right] \quad [U]$$

$$(U_{i-1,j} - U_{i,j}) \left[\frac{\pi \sin(\phi_i - \phi_{i-1}) (r_{j+1} - r_{j-1}) \sigma_{i-1/2,j}}{2(\phi_i - \phi_{i-1})} \right] \quad [U]$$

$$(U_{i,j+1} - U_{i,j}) \left[\pi \sigma_{i,j+1/2} (\phi_{i+1} - \phi_{i-1}) \sin \phi_i \frac{(r_{j+1} + r_j)^2}{4(r_{j+1} - r_j)} \right] \quad [U]$$

$$(U_{i,j+1} - U_{i,j}) \left[\pi \sigma_{i,j-1/2} (\phi_{i+1} - \phi_{i-1}) \sin \phi_i \frac{(r_j + r_{j-1})^2}{4(r_j - r_{j-1})} \right] \quad [U]$$

y la fuente :

$$I_{i,j} = J_{i,j} V_{i,j} = I \quad [A]$$

El tratamiento de la colocación de los coeficientes en la matriz a resolver tiene una forma similar al descrito para el caso cilíndrico, así como la multiplicación por el volumen para obtener la matriz pentadiagonal simétrica, pero la formación de las diagonales es diferente debido a las condiciones de frontera y el número de nodos, como puede verse las dimensiones de los coeficientes agrupados son inversas de resistencias y el problema puede ser resuelto como circuito eléctrico.

Se debe hacer una ecuación para cada frontera, y aunque solamente cambian los extremos pues no se tiene la estrella completa modificándose las $i+1$ o $i-1$ para las fronteras del eje y las $j+1, j-1$ para la frontera inferior, así como las dos esquinas, la diagonal principal no tiene cambio:

a) Frontera Inferior en R_1

con límites en los índices:

$$1 < i < i_{\max}; j=0$$

y volumen

$$V(i, 0) = 2\pi r_0^2 \sin \phi_i \left(\frac{r_1 - r_0}{2} \right) \left(\frac{\phi_{i+1} - \phi_{i-1}}{2} \right)$$

$$U_{i+1,0} \left[\frac{2 \sin \phi_{i+1/2} \sigma_{i+1/2,0}}{(\phi_{i+1} - \phi_i)(\phi_{i+1} - \phi_{i-1}) r_0^2 \sin \phi_i} \right]; U_{i-1,j} \left[\frac{2 \sin \phi_{i-1/2} \sigma_{i-1/2,0}}{(\phi_i - \phi_{i-1})(\phi_{i+1} - \phi_{i-1}) r_0^2 \sin \phi_i} \right];$$

$$U_{i,1} \left[\frac{2\sigma_{i,1/2} r_{1/2}^2}{r_0^2 (r_1 - r_0)^2} \right]$$

$$U_{i,j} \left[-\frac{2\sigma_{i,1/2} r_{1/2}^2}{r_0^2 (r_1 - r_0)^2} - \frac{2}{r_0^2 \sin \phi_i (\phi_{i+1} - \phi_{i-1})} \left(\frac{\sigma_{i+1/2,0} \sin \phi_{i+1/2}}{(\phi_{i+1} - \phi_i)} + \frac{\sigma_{i-1/2,0} \sin \phi_{i+1/2}}{(\phi_i - \phi_{i-1})} \right) \right]$$

b) Para la frontera izquierda $\phi = 0^\circ$

y volumen

$$V(0, j) = 2\pi r_j^2 \sin \phi_0 \left(\frac{r_{j+1} - r_{j-1}}{2} \right) \left(\frac{\phi_1 - \phi_0}{2} \right)$$

con índices $i=0, 1 < j < j_{\max}$

$$U_{1,j} \left[\frac{2 \sin \phi_{1/2} \sigma_{1/2,j}}{(\phi_1 - \phi_0)^2 r_j^2 \sin \phi_0} \right]$$

$$U_{0,j+1} \left[\frac{2\sigma_{0,j+1/2} r_{j+1/2}^2}{r_j^2 (r_{j+1} - r_{j-1})(r_{j+1} - r_j)} \right]; U_{0,j-1} \left[\frac{2\sigma_{0,j-1/2} r_{j-1/2}^2}{r_j^2 (r_{j+1} - r_{j-1})(r_j - r_{j-1})} \right]$$

$$U_{0,j} \left[-\frac{2}{r_j^2 (r_{j+1} - r_{j-1})} \left(\frac{r_{j+1/2}^2 \sigma_{0,j+1/2}}{(r_{j+1} - r_j)} + \frac{r_{j-1/2}^2 \sigma_{0,j-1/2}}{(r_j - r_{j-1})} \right) - \frac{2 \sin \phi_{1/2} \sigma_{1/2,j}}{(\phi_1 - \phi_0)^2 r_j^2 \sin \phi_0} \right]$$

c) Para la frontera derecha $\phi = 180^\circ$

y volumen

$$V(i_{\max}, j) = 2\pi r_j^2 \sin \phi_{i_{\max}} \left(\frac{r_{j+1} - r_{j-1}}{2} \right) \left(\frac{\phi_{i_{\max}} - \phi_{i_{\max}-1}}{2} \right)$$

Con índices $i=i_{\max}, 1 < j < j_{\max}$

$$U_{i_{\max},j} \left[\frac{2 \sin \phi_{i_{\max}-1/2} \sigma_{i_{\max}-1/2,j}}{(\phi_{i_{\max}} - \phi_{i_{\max}-1})^2 r_j^2 \sin \phi_{i_{\max}}} \right]$$

$$U_{i_{\max},j+1} \left[\frac{2\sigma_{i_{\max},j+1/2} r_{j+1/2}^2}{r_j^2 (r_{j+1} - r_{j-1}) (r_{j+1} - r_j)} \right]; U_{i_{\max},j-1} \left[\frac{2\sigma_{i_{\max},j-1/2} r_{j-1/2}^2}{r_j^2 (r_{j+1} - r_{j-1}) (r_j - r_{j-1})} \right]$$

$$U_{i_{\max},j} \left[-\frac{2}{r_j^2 (r_{j+1} - r_{j-1})} \left(\frac{r_{j+1/2}^2 \sigma_{i_{\max},j+1/2}}{(r_{j+1} - r_j)} + \frac{r_{j-1/2}^2 \sigma_{i_{\max},j-1/2}}{(r_j - r_{j-1})} \right) \right. \\ \left. - \frac{2 \sin \phi_{i_{\max}-1/2} \sigma_{i_{\max}-1/2,j}}{(\phi_{i_{\max}} - \phi_{i_{\max}-1})^2 r_j^2 \sin \phi_{i_{\max}}} \right]$$

d) La evaluación de los potenciales se hace en las esquinas, así que para la esquina izquierda con índices $i=0, j=0$ y con la unión de las condiciones de frontera de ambos lados, el lado perteneciente al radio interno y el lado izquierdo adyacente al eje de simetría

$$U_{0,1} \left[\frac{2\sigma_{0,1/2} r_{1/2}^2}{r_0^2 (r_1 - r_0)^2} \right]$$

$$U_{1,0} \left[\frac{2 \sin \phi_{1/2} \sigma_{1/2,0}}{(\phi_1 - \phi_0)^2 r_0^2 \sin \phi_0} \right]$$

$$U_{0,0} \left[-\frac{2\sigma_{0,1/2} r_{1/2}^2}{r_0^2 (r_1 - r_0)^2} - \frac{2 \sin \phi_{1/2} \sigma_{1/2,0}}{(\phi_1 - \phi_0)^2 r_0^2 \sin \phi_0} \right]$$

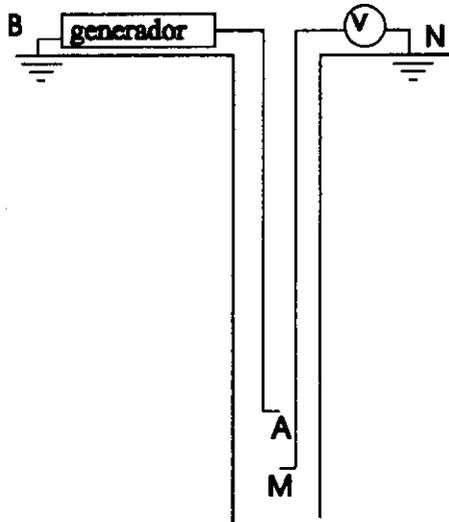
Para la esquina derecha $j=0, i=i_{\max}$

$$U_{i_{\max},1} \left[\frac{2\sigma_{i_{\max},1/2} r_{1/2}^2}{r_0^2 (r_1 - r_0)^2} \right]$$

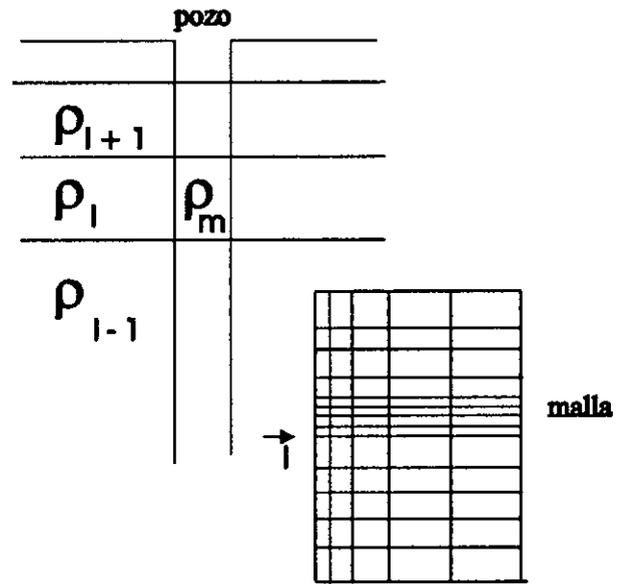
$$U_{i_{\max},0} \left[\frac{2 \sin \phi_{i_{\max}-1/2} \sigma_{i_{\max}-1/2,0}}{(\phi_{i_{\max}} - \phi_{i_{\max}-1})^2 r_0^2 \sin \phi_{i_{\max}}} \right]$$

$$U_{i_{\max},0} \left[-\frac{2\sigma_{i_{\max},1/2} r_{1/2}^2}{r_0^2 (r_1 - r_0)^2} - \frac{2 \sin \phi_{i_{\max}-1/2} \sigma_{i_{\max}-1/2,0}}{(\phi_{i_{\max}} - \phi_{i_{\max}-1})^2 r_0^2 \sin \phi_{i_{\max}}} \right]$$

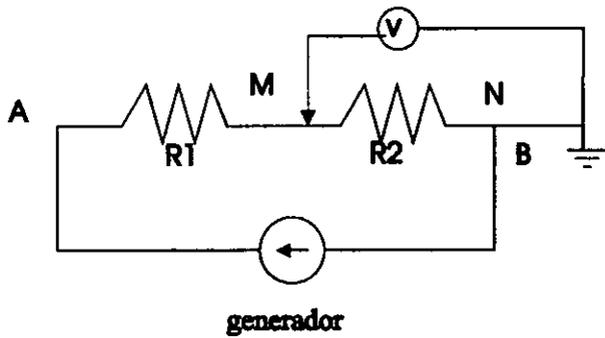
La evaluación del potencial con diferencias finitas esféricas requiere como en el caso cilíndrico multiplicar por el volumen de la celda al nodo de corriente, pero como se indicó en el caso cilíndrico al multiplicar por el volumen se cancela con el volumen mismo de la celda, dependiendo únicamente de la intensidad de corriente I .



instrumento normal - disposición básica



malla y medio coincidente



instrumento normal - circuito

Figura 1.1: Herramienta normal corta y malla cilíndrica

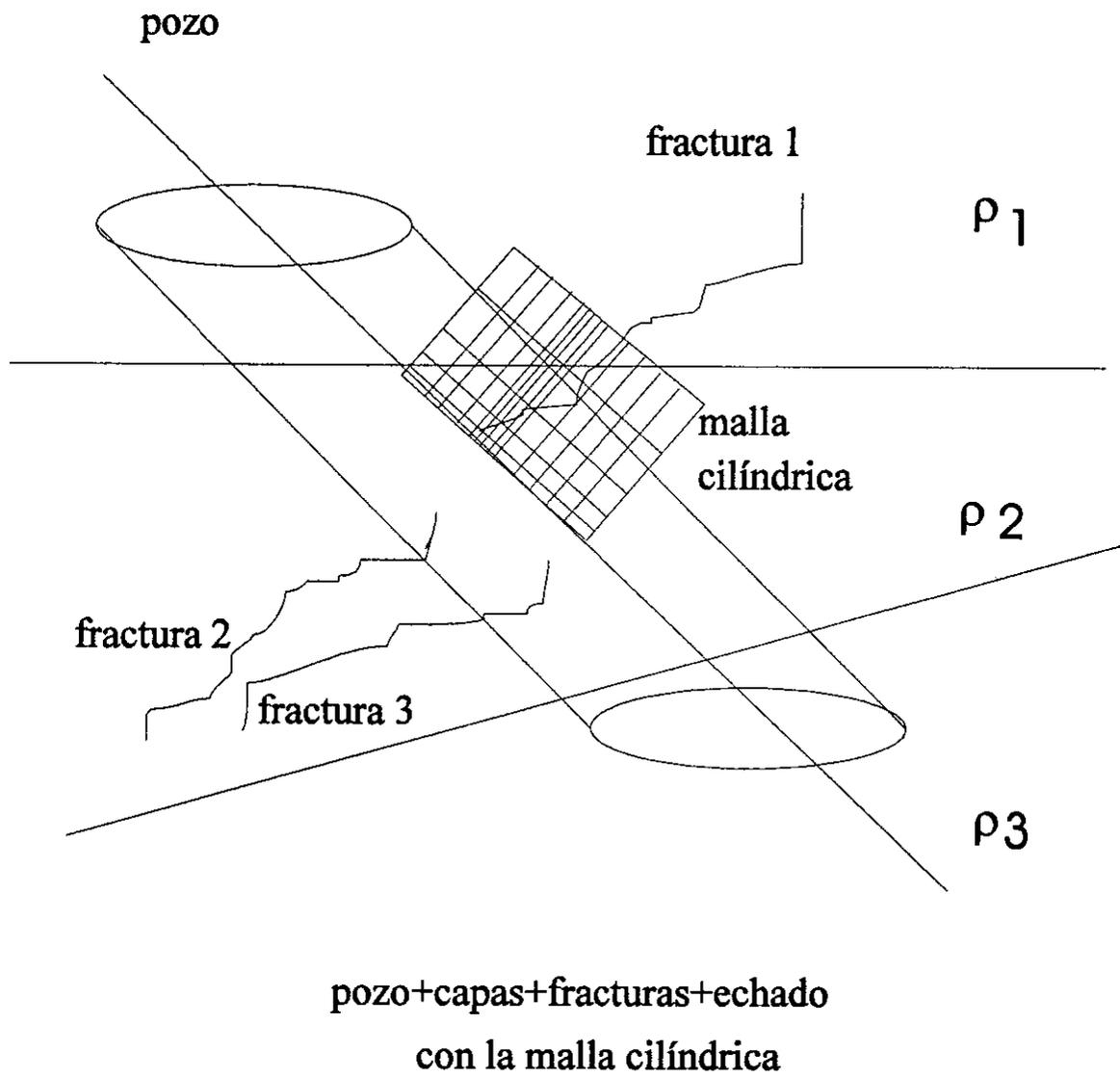


Figura 1.2: Malla cilíndrica en medios no coincidentes

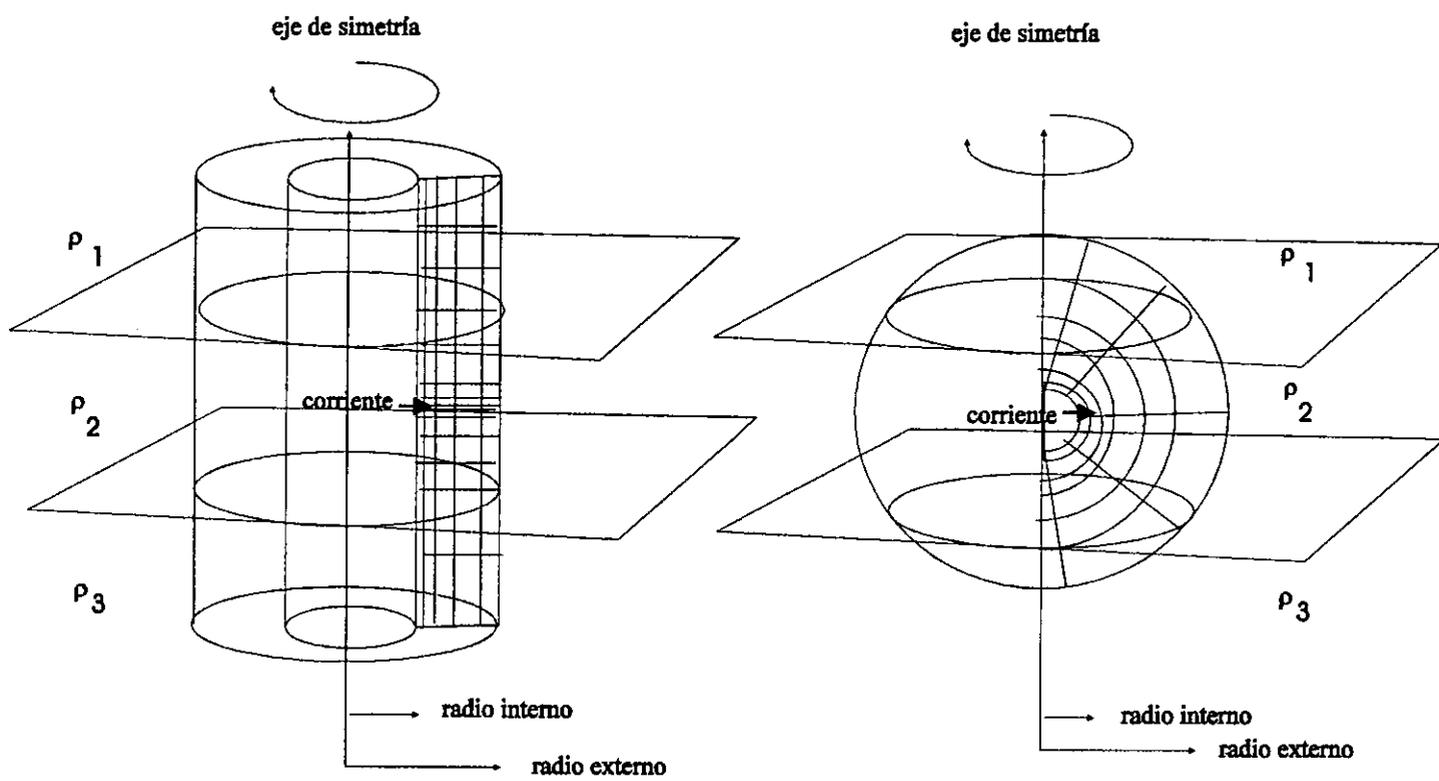


Figura 1.3: Mallas cilíndrica y esférica

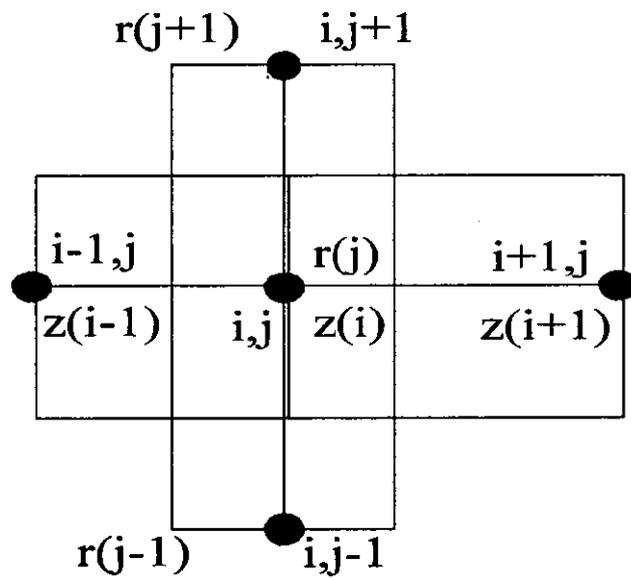


Figura 1.4: Sistema de Discretización

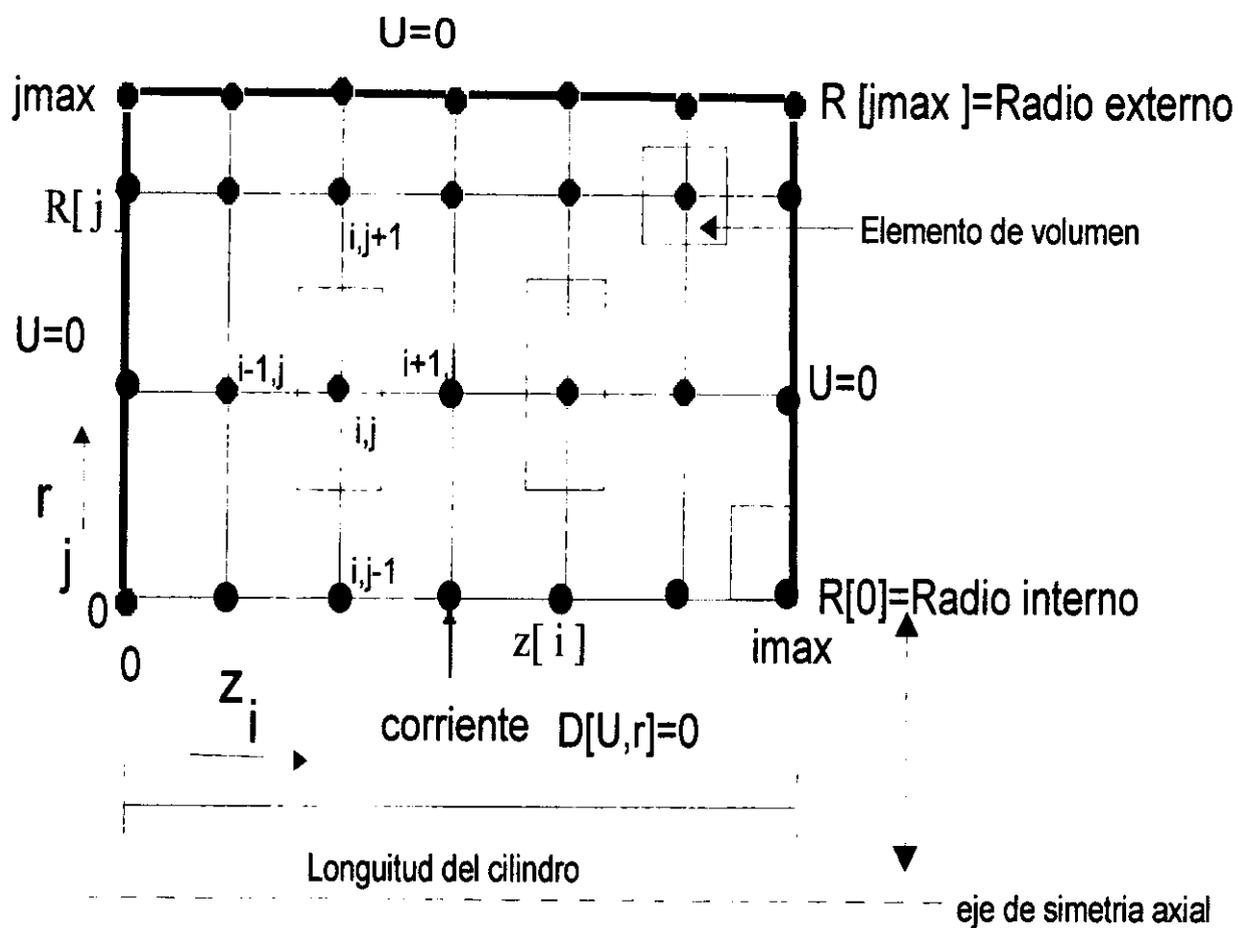


Figura 1.5: Malla de Diferencias Finitas Cilíndrica

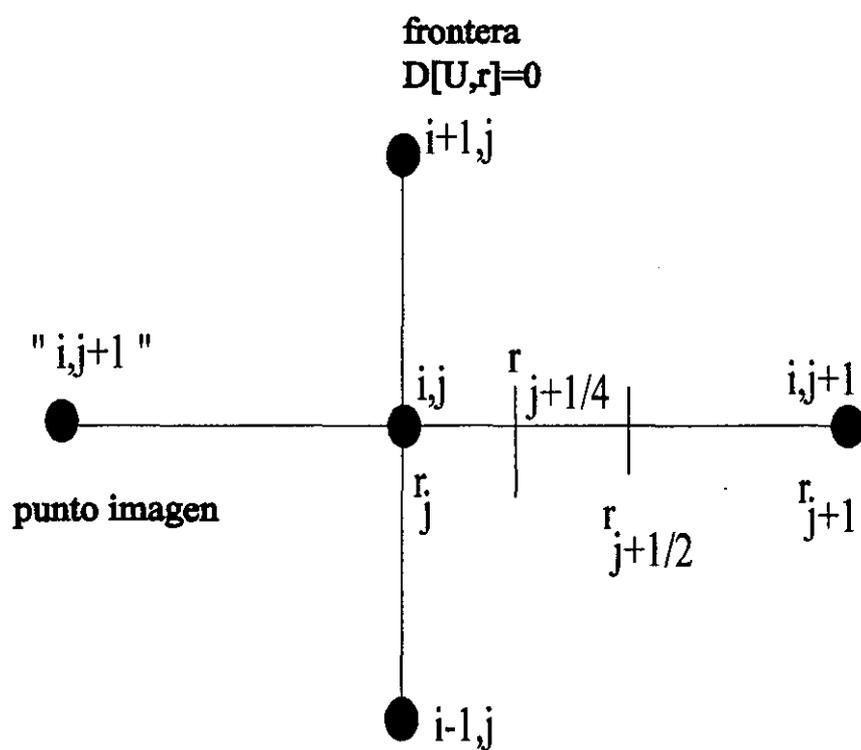


Figura 1.6: Derivación en la frontera

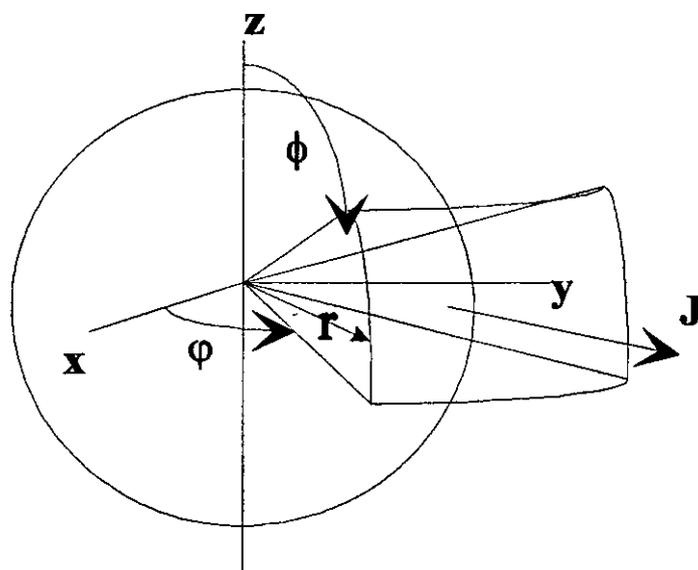


Figura 1.7: Sistema Esférico

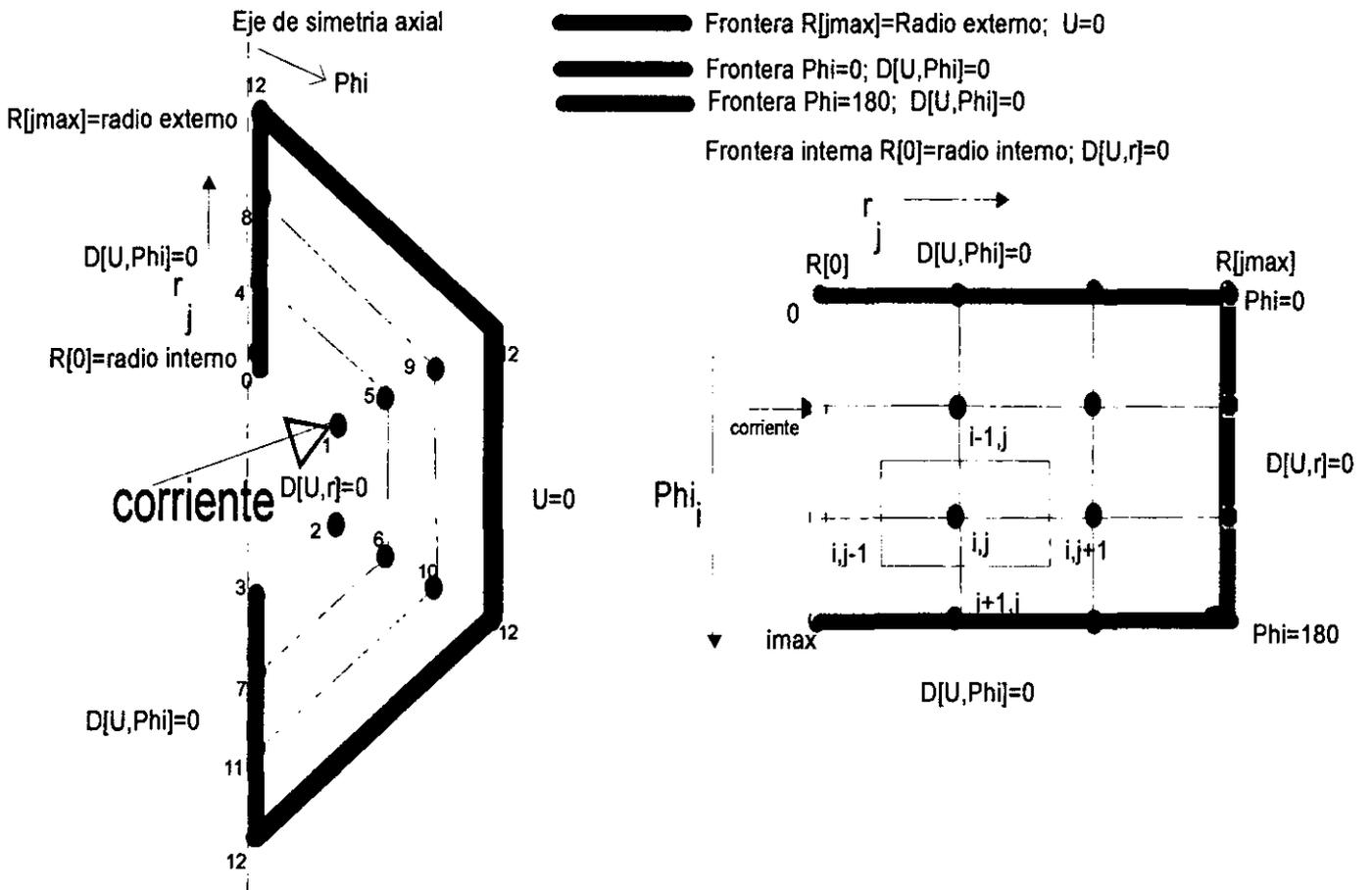


Figura 1.8: Malla Esférica

Capítulo 2

Red de Resistencias Eléctricas

2.1 Introducción

El campo eléctrico puede ser evaluado utilizando analogías de circuitos, que modela la distribución de la resistividad terrestre. La analogía con circuitos fue concebida primeramente en los laboratorios del Instituto del Petróleo de Moscú por E.M Hubkin en 1953. En la actualidad el problema de determinar la resistividad aparente con analogías de circuitos esta bien estudiada.

La analogía de resistencias consiste en hacer construir una red de resistencias o circuito eléctrico que tienen la resistividad del medio y que tienen las dimensiones de acuerdo a la geometría utilizada, una vez enlazadas se colocan las fuentes de corriente en alguno de los nodos de la malla y se obtienen los potenciales nodales, debido a estas fuentes respecto a una tierra virtual a la que se le asigna el cero absoluto, la distancia a la que se coloca la tierra es una parámetro importante en la construcción de la malla así como las dimensiones de la celda.

Existen dos formas de resolver el circuito, ambos se basan en las leyes de Kirchhoff de corrientes y voltajes, pero uno se trabaja con la evaluación nodal de la corrientes mientras el segundo utiliza la teoría matricial de circuitos, los dos llegan a una matriz pentadiagonal simétrica, la teoría matricial no hace mejoras en la rapidez de resolución del circuito solamente ofrece la posibilidad de visualizar la formación de la matriz pentadiagonal así como de jugar con diferentes topologías.

2.2 El Concepto de Resistencia

Consideraremos un elemento arbitrario de tubo de corriente C , que está confinado por dos secciones cruzadas $S1$ y $S2$ y la superficie lateral SL , como en la figura 2.1:

La diferencia de potencial o el voltaje, entre dos superficies equipotenciales $S1$ y $S2$ es:

$$V_{12} = U(1) - U(2) = \int_1^2 \mathbf{E} \cdot d\mathbf{L} \quad (2.1)$$

o

$$V_{12} = U(1) - U(2) = \int_1^2 \rho \mathbf{j} \cdot d\mathbf{L}$$

Esto es obvio, ya que el campo eléctrico es el campo origen y la integral es independiente de la trayectoria de integración, se puede integrar a lo largo de cualquier trayectoria L teniendo los puntos terminales en las superficies $S1$ y $S2$. Nótese que a lo largo de estas líneas de corrientes L son largos y pasan a través de partes más resistivas del medio, la magnitud de la densidad de corriente es más pequeña que a lo largo de líneas cortas que pasan por medios conductores. En otras palabras, en general la densidad de corriente no está uniformemente distribuida en la sección transversal del tubo de corriente.

La ecuación se puede representar como

$$U(1) - U(2) = IR_{12}$$

dónde

$$R_{12} = \int_1^2 \rho \frac{\mathbf{j} \cdot d\mathbf{L}}{I}$$

R_{12} es llamada resistencia del elemento del tubo de corriente C , cuando la corriente va de la superficie $S1$ a $S2$. Es claro que las dimensiones de R son

$$[R] = ohm$$

Ahora haremos varios comentarios, refiriendonos al concepto de resistencia.

1. La resistencia es siempre positiva, y tiene la trayectoria de integración L , mostrando la dirección de S_1 a S_2 , asumimos que $I > 0$. Si escogemos la dirección opuesta de L entonces es necesario cambiar el signo de la corriente.

2. La resistencia R_{12} definida arriba, tiene sentido físico sólo si ambas superficies S_1 y S_2 son superficies equipotenciales y SL es la superficie lateral del tubo de corriente. En efecto supongamos que la superficie S no es una superficie equipotencial. Entonces cambiando la posición de un punto "1" en esta superficie, el voltaje y por ende la resistencia del conductor C cambiará, y R_{12} no tendría sentido. Esto pasa también si la superficie SL no es una lateral, entonces en este caso la corriente I depende de la sección transversal S del conductor, mientras que en la ecuación de arriba se asume que I es constante.

3. La resistencia R_{12} caracteriza la habilidad del conductor C de resistir el paso de la corriente. Esto es numéricamente igual a la diferencia de potencial $U(1)-U(2)$ teniendo la corriente un valor unitario.

4. Para determinar los parámetros que definen la resistencia R_{12} , se hará uso del teorema de unicidad, asumiendo que el conductor es uniforme. Sin embargo, después se demostrará que esto es válido en cualquier caso, donde se tienen cambios arbitrarios de resistividad en un conductor. Es claro que el potencial del campo eléctrico satisface las siguientes condiciones:

(a) U es una solución a la ecuación de Laplace dentro de un conductor homogéneo

$$\nabla^2 U = 0$$

(b) En la superficie lateral SL , la componente normal del campo eléctrico es CERO, o sea:

$$\frac{\partial U}{\partial n} = 0 \text{ en } SL$$

(c) En las superficies equipotenciales S_1 y S_2 , el potencial cumple con:

$$\frac{1}{\rho} \int_{S_1} \frac{\partial U}{\partial n} dS = -I, \quad \frac{1}{\rho} \int_{S_2} \frac{\partial U}{\partial n} dS = -I,$$

Según el teorema de unicidad, estas tres condiciones definen el campo eléctrico dentro del conductor C , en otras palabras, sólo puede existir una distribución de

campo si la corriente es dada. Supongamos una corriente I_1 y su correspondiente campo eléctrico \mathbf{E}_1 , que por supuesto satisface las tres condiciones. Ahora incrementemos la corriente m veces, esto es $I_2 = mI_1$, es fácil ver que esta carga da como resultado un incremento de la magnitud del campo eléctrico por un factor de m en cada punto del conductor, pero "no cambia de dirección". El nuevo campo es:

$$\mathbf{E}_2 = m\mathbf{E}_1$$

y el potencial del campo de \mathbf{E}_2 será

$$U_2 = mU_1$$

Siendo U_1 y U_2 soluciones a la ecuación de Laplace. Además este campo cumple al mismo tiempo las condiciones de frontera de las superficies transversales y de la lateral :

$$\frac{\partial U_2}{\partial n} = m \frac{\partial U_1}{\partial n}$$

Entonces, de acuerdo con el teorema de unicidad, podemos decir que la función \mathbf{E}_2 describe el campo eléctrico en el conductor C si la corriente que atraviesa la sección transversal es igual a I_2 . Este análisis muestra que las relaciones

$$\frac{\mathbf{E}}{I} \text{ o } \frac{\rho \mathbf{j}}{I}$$

son independientes de la magnitud de la corriente. En otras palabras, el campo vectorial \mathbf{j}/I en el volumen V está enteramente determinado por sus fronteras S_1 , S_2 y S_L y de la distribución de la resistividad ρ como en la figura 2.2.

Entonces la resistencia R de un conductor C queda definida por sus dimensiones, forma y resistividad, no depende de la corriente o carga de cualquier otra parte del circuito si las superficies S_1 y S_2 son equipotenciales. Por supuesto que para definir la resistencia R_{12} , implicamos sólo una dirección en la que se introduce corriente en un circuito eléctrico para conservar las superficies equipotenciales.

5. Para determinar la resistencia es necesario conocer el campo densidad de corriente, que en general es difícil en problemas con valores en la frontera.

6. Otras formas de la ecuación de ohm son:

$$\mathbf{j} = \sigma \mathbf{E} \text{ y } U(1) - U(2) = R_{12} I$$

7. Si la magnitud del vector \mathbf{j} tiene iguales valores, en cada sección del conductor, perpendicular a la corriente, entonces tenemos:

$$\frac{\mathbf{j} \cdot d\mathbf{L}}{I} = \frac{j dL}{I} = \frac{dL}{S}$$

y

$$R_{12} = \int_1^2 \rho \frac{dL}{S} \quad (2.2)$$

En particular si el conductor es uniforme:

$$R_{12} = \rho \int_1^2 \frac{dL}{S} \quad (2.3)$$

Es importante notar que las ecuaciones 2.2 y 2.3 pueden ser utilizadas sólo si se conoce la geometría del campo densidad de corriente, ya que S son secciones de un conductor que son perpendiculares al vector \mathbf{j} .

8. Supongamos que el conductor C es un cilindro de tamaño arbitrario, confinado por una superficie lateral y dos superficies equipotenciales transversales S_1 y S_2 , como en la figura 2.3

En este caso el campo eléctrico se comportará notablemente simple, será constante en el conductor y paralelo a la superficie lateral, entonces el potencial del campo será una función lineal y satisface la ecuación de Laplace con la condición de frontera:

$$\frac{\partial U}{\partial n} = 0 \text{ en SL}$$

y

$$\sigma \int_{S_1} \frac{\partial U}{\partial n} dS = \sigma \int_{S_2} \frac{\partial U}{\partial n} dS$$

Siendo el campo eléctrico uniforme y el vector densidad de corriente también, la expresión de resistencia se simplifica a:

$$R_{12} = \rho \frac{L}{S}$$

Donde L y S son la longitud y el área transversal del cilindro respectivamente.

El campo eléctrico dentro de un cilindro uniforme es causado por cargas en la superficie que son distribuidas de tal manera que los campos densidad y eléctrico son uniformes y paralelos a la superficie transversal.

9. Es claro que si la resistividad de un cilindro ρ , no cambia dentro cada sección transversal pero varía a lo largo del conductor, entonces el campo densidad de corriente j será uniforme y paralelo a la superficie lateral. Al mismo tiempo el campo eléctrico cambia a lo largo del cilindro pero es constante en las secciones y directamente proporcional a la resistividad.

$$E = \rho j = \rho \frac{I}{S}$$

Teniendo también la expresión de resistencia de un cilindro conductor confinado por las superficies equipotenciales S_1 y S_2 .

$$R_{12} = \frac{1}{S} \int_1^2 \rho dL$$

10. Se ha hecho énfasis en que la resistencia en un conductor C solo tiene sentido si las superficies S_1 y S_2 son superficies equipotenciales sin tener en cuenta otras partes del circuito. Al mismo tiempo es muy simple hacer modificaciones fuera del conductor C que resulta en un cambio de potencial sobre las superficies S_1 y S_2 o en una de ellas.

Si se introduce una inhomogeneidad T , las cargas eléctricas en la superficie crean un campo que ciertamente no es perpendicular a la superficie S_2 . Es también claro que cerca de la inhomogeneidad y de la superficie S_2 su influencia es grande. Sin embargo hay casos en que las superficies S_1 y S_2 pueden ser solamente equipotenciales. Entonces esto se cumple si estas superficies son parte de un conductor ideal o una sección transversal de una parte lineal de un circuito.

2.3 Teoría de circuitos

Una vez establecidas las resistencias eléctricas es necesario establecer un método de solución del circuito eléctrico, tradicionalmente se utilizan las ecuaciones de Kirchhoff de voltaje y corriente. Resolviendo para los potenciales se forma un sistema de ecuaciones para cada nodo de la malla, en la que los coeficientes son las resistencias adyacentes a dicho nodo, la matriz a invertir tiene entonces la forma pentadiagonal simétrica positiva:

$$\begin{bmatrix} d_0 & -a_0 & 0 & -c_0 & 0 & 0 & 0 & 0 \\ -a_0 & d_1 & -a_1 & 0 & -c_1 & 0 & 0 & 0 \\ 0 & -a_1 & d_2 & 0 & 0 & -c_2 & 0 & 0 \\ -c_0 & 0 & 0 & d_3 & -a_2 & 0 & -c_3 & 0 \\ 0 & -c_1 & 0 & -a_2 & d_4 & -a_3 & 0 & \cdot \\ 0 & 0 & -c_2 & 0 & -a_3 & d_5 & \cdot & 0 \\ 0 & 0 & 0 & \cdot & 0 & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & -c_{i-1} & 0 & -a_{i-1} & d_i \end{bmatrix}$$

La teoría de circuitos utiliza los métodos matriciales para encontrar las relaciones entre elementos del circuito y los parámetros a determinar.

2.4 Uso de matrices para expresar las leyes de Kirchhoff y la relaciones de lazo.

Ley de Corrientes de Kirchhoff (LCK)

Para expresar las $n_{nodal} - 1$ ecuaciones linealmente independientes de las ecuaciones nodales para la ley de Kirchhoff de corrientes en términos de los n_b corrientes de lazo se tiene:

$$\sum_{k=1}^{n_b} a_{jk} i_k = 0 \text{ para } j=1,2,\dots,n_n - 1$$

donde los coeficientes a_{jk} son definidos como sigue:

$$a_{jk} = \begin{cases} +1 & \text{si en el lazo } k \text{ sale la corriente del nodo } j \\ -1 & \text{si en el lazo } k \text{ entra la corriente al nodo } j \\ 0 & \text{si el lazo } k \text{ no tiene contacto con el nodo } j \end{cases}$$

Entonces se define la matriz de incidencia nodal \mathbf{A} con coeficientes a_{jk} .

$$\mathbf{A} = [a_{jk}]$$

Y el vector de corrientes de lazo

$$\mathbf{i}_b = \begin{bmatrix} i_1 \\ i_2 \\ \cdot \\ \cdot \\ \cdot \\ i_{nb} \end{bmatrix}$$

La ecuación de corrientes de kirchhoff toma la forma

$$\mathbf{A}\mathbf{i}_b = \mathbf{0} \quad (2.4)$$

Siendo \mathbf{A} la matriz incidencia del circuito que juega un papel muy importante en la teoría de circuitos y se define como la matriz de $n_n \times n_b$ siendo n_n número de nodos y n_b número de lazos en una malla.

La formación de la matriz incidencia es un proceso sencillo, la primera dimensión de la matriz es el número de nodos de la malla y la segunda el número de lazos. Cada lazo está relacionado con dos nodos a los que se le asignan polaridades +1 y -1, de tal forma que al terminar de formar la matriz \mathbf{A} en cada columna la suma sea cero. La matriz de incidencia formada de esta forma debe ser reducida en un renglón para obtener la independencia lineal así que se toma un nodo de referencia a tierra.

El proceso puede describirse:

1) Formación de la matriz incidencia (conocida también como matriz primitiva pues no tiene ningún nodo de referencia a tierra, esto es que está suspendida en

el espacio), cada columna es un lazo del circuito, que esta relacionado con unicamente dos nodos o dos renglones.

2) Eliminación del renglón o nodo de referencia, que origina a la matriz de incidencia final.

La ley de voltajes de Kirchhoff

Utilizando la matriz incidencia, la ley de voltajes de Kirchhoff (LVK) queda:

$$\mathbf{v}_b = \mathbf{A}^T \mathbf{v}_n \quad (2.5)$$

\mathbf{v}_b es el vector de voltajes de lazo

\mathbf{v}_n es el vector de voltajes de nodo

Además, las relaciones de lazo: Si el lazo k -ésimo ($k = 1, 2, \dots, n_b$) es una resistencia R_k entonces se pueden expresar las relaciones de lazo como:

$$i_k = \left(\frac{1}{R_k} \right) v_k$$

por otro lado si el lazo k -ésimo es una fuente de corriente de valor \hat{i}_k entonces:

$$i_k = \hat{i}_k$$

Todas las relaciones de lazo pueden ser expresadas en forma matricial como:

$$\mathbf{i}_b = \mathbf{Z} \mathbf{v}_b + \mathbf{i} \quad (2.6)$$

donde $\mathbf{Z} = [z_{jk}]$ es una matriz cuadrada de orden $n_b \times n_b$, siendo que $z_{jk} = 0$ para $j \neq k$ y

$$z_{kk} = \begin{cases} 1/R_k & \text{si en el lazo } k\text{-ésimo es una resistencia de valor } R_k \\ 0 & \text{si el lazo } k\text{-ésimo es una fuente de corriente} \end{cases}$$

y $\mathbf{i} = [i_k]$ es un vector de orden n_b tal que:

$$i_k = \begin{cases} 0 & \text{si en el lazo } k\text{-ésimo es una resistencia} \\ \hat{i}_k & \text{si el lazo } k\text{-ésimo es una fuente de corriente} \end{cases}$$

Sustituyendo 2.4 en 2.6:

$$\mathbf{A} (\mathbf{Z}\mathbf{v}_b + \mathbf{i}) = \mathbf{0}$$

$$\mathbf{AZ}\mathbf{v}_b + \mathbf{Ai} = \mathbf{0}$$

pero $\mathbf{v}_b = \mathbf{A}^T \mathbf{v}_n$

$$\mathbf{AZA}^T \mathbf{v}_n + \mathbf{Ai} = \mathbf{0}$$

o

$$\mathbf{AZA}^T \mathbf{v}_n = -\mathbf{Ai}$$

La matriz incidencia permite jugar con la topología del circuito y modelar de cualquier forma para poder observar la forma en que se crea la matriz de coeficientes y programar el algoritmo para casos generales.

La matriz \mathbf{AZA}^T

Como se describió anteriormente la matriz \mathbf{AZA}^T es pentadiagona, simétrica y su inversión es una parte muy importante en la solución del problema, la rapidez con que se consiga resolver el sistema de ecuaciones tendrá repercusiones en la efectividad de la inversión.

Existen diversos algoritmos que resuelven el problema y se basan en la propiedad de la matriz de ser simétrica positiva, que la hace estable y puede ser resuelta por el algoritmo de Choleski o Choleski vandeado.

Pero la teoría matricial de circuitos podría ofrecer otra forma de resolver el sistema de ecuaciones, por la forma clásica es necesario realizar la inversión de toda la matriz cada vez que se muestrea cada punto, pero si de alguna forma se lograra convertir la matriz incidencia que por naturaleza es singular, en una matriz no singular, entonces el proceso se reduciría a obtener la inversa de la matriz incidencia solamente un vez para todos los muestreos, esto sería:

\mathbf{A} no singular

$$\mathbf{A}^{-1} \mathbf{AZA}^T \mathbf{v}_n = -\mathbf{A}^{-1} \mathbf{Ai}$$

$$\mathbf{Z}\mathbf{A}^T\mathbf{v}_n = -\mathbf{i}$$

$$\mathbf{v}_n = -(\mathbf{Z}\mathbf{A}^T)^{-1}\mathbf{i}$$

$$\mathbf{v}_n = -\mathbf{Z}^{-1}(\mathbf{A}^T)^{-1}\mathbf{i}$$

pero \mathbf{Z} es diagonal y $\mathbf{Z}^{-1} = 1/Z$, convertiría el proceso de uno de los más rápidos existentes.

Poblemas con la matriz \mathbf{A} no singular

Por naturaleza la matriz incidencia \mathbf{A} es no cuadrada y singular, pues siempre se tienen más lazos que nodos en el circuito y estos son las dimensiones de la matriz, así que el único camino para transformarla en no singular es la teoría de circuitos. En la teoría de circuitos se establece que si se tiene una matriz \mathbf{B} de circuito definida

$$\mathbf{B} = -[\mathbf{A}_{12}^{-1} \cdot \mathbf{A}_{11}]^T$$

Donde la matriz incidencia es dividida en dos partes

$$\mathbf{A} = [\mathbf{A}_{11} \mathbf{A}_{12}]$$

siendo

\mathbf{A}_{11} = elementos en serie del circuito

y

\mathbf{A}_{12} = elementos en paralelo

Entonces

$$\mathbf{A} \cdot \mathbf{B}^T = 0$$

que dice que \mathbf{B} es una matriz ortogonal a \mathbf{A} , sugiriendo posible establecer

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{Z}[\mathbf{A}\mathbf{B}] = - \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{i}$$

Por otra parte la matriz \mathbf{B} puede ser establecida independientemente de la matriz \mathbf{A} pues se tiene la relación:

$$\mathbf{BZ}^{-1}\mathbf{B}^T\mathbf{i}_b = -\mathbf{B}\mathbf{v}_n$$

que si es \mathbf{B} no singular $\mathbf{B}=[\mathbf{A}\mathbf{B}]$, y resolviendo para \mathbf{v}_n

$$-\mathbf{Z}^{-1}[\mathbf{A}\mathbf{B}]^T\mathbf{i}_b = \mathbf{v}_n \quad (2.7)$$

que junto con $\mathbf{A} = \begin{bmatrix} A \\ B \end{bmatrix}$

$$-\mathbf{Z}^{-1}([\mathbf{A}\mathbf{B}]^T)^{-1}\mathbf{i} = \mathbf{v}_n \quad (2.8)$$

Entonces 2.7 queda

$$-\mathbf{Z}^{-1} \begin{bmatrix} A \\ B \end{bmatrix} \mathbf{i} = \mathbf{v}_n$$

y si $([\mathbf{A}\mathbf{B}]^T)^{-1} = \begin{bmatrix} A \\ B \end{bmatrix}$ 2.8 sería

$$-\mathbf{Z}^{-1} \begin{bmatrix} A \\ B \end{bmatrix} \mathbf{i} = \mathbf{v}_n$$

Las matrices incidencia y de circuito forman relaciones que permiten el análisis de los elementos de un circuito y así obtener corrientes o voltajes con ellas, setienen así en forma general con variación en el tiempo y utilizando la notación de Laplace:

A) Las ecuaciones primarias (postulados)

$$\text{LCK: } \mathbf{A}\mathbf{i}_b(s) = 0$$

$$\text{LVK: } \mathbf{B}\mathbf{v}_b(s) = 0$$

RVC (Relación Corrientes Voltajes): $\mathbf{v}_b(s) = \mathbf{Z}\mathbf{e}(s)\mathbf{i}_b(s) + \mathbf{E}_b(s) + \frac{1}{s}\mathbf{V}_b(0^+) - \mathbf{L}_b\mathbf{i}_b$
donde $\frac{1}{s}\mathbf{V}_b(0^+)$, representan los elementos capacitivos y $\mathbf{L}_b\mathbf{i}_b(0^+)$ a los

inductivos.

B) Ecuaciones secundarias

Relación de malla: $\mathbf{i}_b = \mathbf{B}^T \mathbf{i}_m$

Relación de nodos: $\mathbf{v}_b = \mathbf{A}^T \mathbf{v}_n(s)$

Para un análisis de mallas

$$\mathbf{BZ}_b \mathbf{i}_b = -\mathbf{B} \left(\mathbf{E}_b(s) + \frac{1}{s} \mathbf{V}_b(0^+) - \mathbf{L}_b \mathbf{i}_b(0^+) \right)$$

$$\mathbf{BZ}_b \mathbf{B}^T \mathbf{i}_m = -\mathbf{B} \left(\mathbf{E}_b(s) + \frac{1}{s} \mathbf{V}_b(0^+) - \mathbf{L}_b \mathbf{i}_b(0^+) \right)$$

Si no hay variación en el tiempo se tendrían solamente elementos pasivos y no existen inductancias mutuas ni capacitores, \mathbf{Z}_b es una matriz diagonal de conductividades, \mathbf{E}_b es el vector de fuentes de voltaje por lazo, \mathbf{i}_m es el vector incógnita de corrientes de malla reduciéndose a

$$\mathbf{BZ}_b \mathbf{B}^T \mathbf{i}_m = -\mathbf{B} \mathbf{E}_b$$

Por el análisis de nodos $\mathbf{Z} = \mathbf{Z}_b^{-1}$

$$\mathbf{AZ}_b^{-1} \mathbf{A}^T \mathbf{v}_n = -\mathbf{A} \mathbf{I}$$

Siendo \mathbf{A} la matriz incidencia y \mathbf{B} la de circuito, con el mismo orden de columnas:

$$\mathbf{AB}^T = \mathbf{0}$$

o

$$\mathbf{BA}^T = \mathbf{0}$$

El método antes descrito de encontrar solo una inversa podría ser utilizado si se encuentra la matriz de circuito adecuada que satisfaga las ecuaciones, pero en la práctica la búsqueda de la matriz ortogonal \mathbf{B} no es sencilla pues no es única.

2.5 La Teoría Matricial y Ley de Corrientes de Kirchhoff

Como se mencionó anteriormente el problema de determinar el potencial con la analogía de resistencias ya ha sido resuelto y con buenos resultados, este método actualmente se utiliza en forma regular en la interpretación de registros eléctricos de pozos y también ha sido utilizado para investigación de las propiedades de medición de las herramientas como ha sido el caso de la descomposición de los efectos de medición de la lateral en la normal y su derivada, la investigación entonces se convierte en el punto de partida de esta tesis; esto es, no sólo modelar el medio con un método numérico y obtener una curva de resistividad aparente que refleja las propiedades, sino también encontrar los parámetros que controlan al algoritmo de modelado. El programa actual se basa en la ley de corrientes de Kirchhoff nodal, así que evalúa para cada nodo de la malla colocando en la matriz pentadiagonal a resolver los coeficientes de dicha ley, toma en cuenta las condiciones de frontera y resuelve el potencial dependiendo del contraste de resistividad, para entonces obtener la curva de resistividad aparente, como la geometría del pozo es coincidente con la de la malla que trabaja el problema, resolver para cambios horizontales y verticales de resistividad no exige problema. Este programa puede ser utilizado para trabajar con medios no coincidentes con la malla, se aplica entonces una transformación de coordenadas con el jacobiano de transformación y el problema para capas no horizontales queda resuelto y listo para una posterior inversión de datos. La posibilidad de hacer más rápido el modelado directo se basa en encontrar otra forma de resolver el mismo problema y esta otra forma la dan la teoría matricial de circuitos. La comprensión del problema ofrece la posibilidad para poder no sólo modificarlo y entender el problema cilíndrico, sino tener un conocimiento total en 2D para avanzar a 3D y experimentar con una malla esférica que permitiera la disminución de nodos que pudiera acelerar los resultados del problema.

2.6 Resistencias eléctricas con la malla cilíndrica.

El problema se resuelve para un medio homogéneo y el objetivo principal de dicho programa es encontrar el potencial para cualquier punto del espacio con una fuente de corriente y una resistividad constante, entonces se tiene que construir una malla rectangular con las siguientes condiciones de frontera:

Considerando el eje del pozo como aislante de corrientes no existen corrientes hacia el eje del pozo excepto en la fuente de corriente, las otras tres fronteras se colocan a un potencial cero o tierra, la corriente emanada por la fuente de corriente colocada en cualquier parte de la frontera izquierda del rectángulo (eje del cilindro o pozo al rotar la retícula 360°) seguirá el camino hacia los lados de potencial a tierra, lo siguiente es establecer las resistencias apropiadas al medio, se comenzó este trabajo con espacios iguales de distancia en r y z de tal forma que la malla fuese cuadrículada homogéneamente. A continuación se construyen las resistencias de este medio que forman el circuito a resolver, para esto se utilizan las ecuaciones desarrolladas por Daknov para geometría cilíndrica y se basan en considerar el vector densidad de corriente perpendicular a dos caras de un elemento de volumen cilíndrico, primero cuando la corriente atraviesa dicho volumen en dirección radial y luego en dirección vertical. Cuando la corriente atraviesa perpendicularmente el elemento de área perpendicular a la coordenada radial y utilizando la ecuación general de resistencia y la figura 2.4:

$$R = \rho \frac{dL}{dA}$$

Donde

dL = diferencial de longitud

dA = diferencial de área perpendicular al vector densidad de corriente \mathbf{J}

De aquí se tiene para la resistencia radial que

$$dA = r d\phi dz; dL = dr$$

ó sea

$$RdA = \rho dL$$

con límites :

$$R \int_0^{2\pi} \int_{z_1}^{z_2} r d\phi dz = \rho \int_{r_1}^{r_2} dr$$

$$R = \frac{\rho}{2\pi(z_2 - z_1)} \ln \frac{r_2}{r_1}$$

En notación nodal i, j

$$R = \frac{\rho}{2\pi(z_{i+1} - z_i)} \ln \frac{r_{j+1}}{r_j}$$

Para las resistencias verticales en la dirección z

$$R \int_0^{2\pi} \int_{r_1}^{r_2} r d\phi dr = \rho \int_{z_1}^{z_2} dz$$

$$R = \frac{\rho(z_2 - z_1)}{2\pi(r_2^2 - r_1^2)}$$

en notación i, j:

$$R = \frac{\rho(z_{i+1} - z_i)}{2\pi(r_{j+1}^2 - r_j^2)}$$

Que en la ecuación de la figura 2.5:

$$[\mathbf{A} \cdot \mathbf{Z} \cdot \mathbf{A}^T] \cdot \mathbf{v} = -\mathbf{A} \cdot \mathbf{i}_b$$

La matriz incidencia \mathbf{A} del circuito

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & -1 & -1 & 0 & -1 & -1 & 0 & -1 \end{bmatrix}$$

$$\mathbf{i}_b = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]$$

$$\mathbf{v} = [v_0, v_1, v_2, v_3, v_4, v_5]$$

D es la matriz diagonal formada por dos partes, la superior son las resistencias en la dirección radial, mientras que la segunda parte son las resistencias en la dirección z:

En forma general

$$\begin{bmatrix} 1/R_0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cdot & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdot & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/R_n & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0_{(\text{lazo de corriente})} \end{bmatrix}$$

La matriz de conductancias $[\mathbf{A} \cdot \mathbf{Z} \cdot \mathbf{A}^T]$ para el circuito con ancho de banda =3

$$\begin{bmatrix} a_1 & -\frac{1}{R_7} & -\frac{1}{R_0} & 0 & 0 & 0 \\ -\frac{1}{R_7} & a_2 & 0 & -\frac{1}{R_1} & 0 & 0 \\ -\frac{1}{R_0} & 0 & a_3 & -\frac{1}{R_{10}} & -\frac{1}{R_2} & 0 \\ 0 & -\frac{1}{R_1} & -\frac{1}{R_{10}} & a_4 & 0 & -\frac{1}{R_3} \\ 0 & 0 & -\frac{1}{R_2} & 0 & a_5 & -\frac{1}{R_{13}} \\ 0 & 0 & 0 & -\frac{1}{R_3} & -\frac{1}{R_{13}} & a_6 \end{bmatrix}$$

donde

$$\begin{aligned} a_1 &= \frac{1}{R_0} + \frac{1}{R_7} + \frac{1}{R_6} \\ a_2 &= \frac{1}{R_1} + \frac{1}{R_7} + \frac{1}{R_8} \\ a_3 &= \frac{1}{R_0} + \frac{1}{R_2} + \frac{1}{R_9} + \frac{1}{R_{10}} \\ a_4 &= \frac{1}{R_3} + \frac{1}{R_{11}} + \frac{1}{R_1} + \frac{1}{R_{10}} \\ a_5 &= \frac{1}{R_4} + \frac{1}{R_{12}} + \frac{1}{R_{13}} + \frac{1}{R_2} \\ a_6 &= \frac{1}{R_5} + \frac{1}{R_{13}} + \frac{1}{R_{14}} + \frac{1}{R_3} \end{aligned}$$

Una matriz de coeficientes toma la forma general

$$\begin{bmatrix} 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 3 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 \end{bmatrix}$$

número de condicionamiento: $(4 + \sqrt{5}) \left(\frac{3}{4} + \frac{1}{4}\sqrt{5}\right) : 8.1631$

El número de condicionamiento de una matriz invertible A es el producto de norma-2 de A y la norma-2 de A^{-1}

$$\text{número de condicionamiento} = \|A\| * \|A^{-1}\|$$

Este número mide la sensibilidad de la solución de una ecuación lineal $Ax=b$ a las perturbaciones o errores de redondeo en A y b, la matriz con número de condicionamiento 1 es perfectamente condicionada.

El número de condicionamiento, indica que tan cerca está la matriz de coeficientes de ser singular, por lo tanto una matriz es singular y no puede invertirse, si su número de condicionamiento es infinito, y es altamente condicionada si el número de condicionamiento es muy grande, esto es, si su recíproco se aproxima a la precisión de punto flotante de la máquina (por ejemplo, menos de 10^{-6} para precisión simple o 10^{-12} para doble precisión) para la malla cilíndrica $1/8.1631 = .1225$, que significa que los problemas con esta malla pueden ser resueltos pues es muy estable y puede ser trabajada con precisión simple.

La forma de construir la matriz de conductancias puede hacerse sin el uso de la matriz incidencia pero ilustra la forma en que se posicionan los elementos en la matriz para formar el caso general en la computadora.

$$-A \cdot \mathbf{i}_b = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

El lado izquierdo de la ecuación o la matriz de conductancias es independiente de la posición y número de fuentes de corriente en la red eléctrica, pues como se vio en la matriz diagonal de conductancias los elementos de corriente son multiplicados por cero.

2.7 Caso esférico

En el caso esférico la malla tiene condiciones de frontera diferentes al caso cilíndrico según la figura 2.6.

En la dirección radial con el sistema de la figura 2.7

$$dA = r^2 \sin \phi d\phi d\theta; dL = dr$$

ó sea

$$RdA = \rho dL$$

con límites :

$$R \int_0^{2\pi} \int_{\phi_1}^{\phi_2} r^2 \sin \phi d\phi d\theta = \rho \int_{r_1}^{r_2} dr$$

$$R = \frac{\rho}{2\pi (-\cos \phi_2 + \cos \phi_1)} \left[\frac{1}{r_1} - \frac{1}{r_2} \right]$$

En notación nodal i, j

$$R = \frac{\rho_{i,j+1/2}}{2\pi (-\cos \phi_{i+1} + \cos \phi_i)} \left[\frac{1}{r_j} - \frac{1}{r_{j+1}} \right]$$

Para las resistencias azimutales en la dirección ϕ

$$R \int_0^{2\pi} \int_{r_1}^{r_2} r \sin \phi d\theta dr = \rho \int_{\phi_1}^{\phi_2} r d\phi$$

$$R = \frac{\rho \left(\ln \tan \frac{\phi_2}{2} - \ln \tan \frac{\phi_1}{2} \right)}{2\pi (r_2 - r_1)}$$

en notación i,j :

$$R = \frac{\rho_{i+1/2,j} \left(\ln \tan \frac{\phi_{i+1}}{2} - \ln \tan \frac{\phi_i}{2} \right)}{2\pi (r_{j+1} - r_j)}$$

Que en la ecuación:

$$[\mathbf{A} \cdot \mathbf{Z} \cdot \mathbf{A}^T] \cdot \mathbf{v} = -\mathbf{A} \cdot \mathbf{i}_b$$

La matriz incidencia \mathbf{A}

$$\begin{array}{c}
 0 \\
 1 \\
 2 \\
 3 \\
 4 \\
 5 \\
 6 \\
 7 \\
 8 \\
 9
 \end{array}
 \left[\begin{array}{cccccccccccccccc}
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & -1 \\
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
 -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\
 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{array} \right]$$

$$\mathbf{i}_b = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1]$$

$$\mathbf{v} = [v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9]$$

\mathbf{Z} es la matriz diagonal formada por dos partes, la superior son las resistencias en la dirección radial, mientras que la segunda parte son las resistencias en la dirección ϕ :

En Forma general

$$\begin{bmatrix} 1/R_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cdot & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdot & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/R_8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/R_9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdot & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdot & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/R_{14} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \text{corriente } 0 \end{bmatrix}$$

Donde $1/R_0 \dots 1/R_8$ son las conductancias en serie y $1/R_9 \dots 1/R_{14}$ son las resistencias en paralelo,

el último elemento de la diagonal es cero debido a que pertenece al lazo donde esta la corriente.

La matriz de conductancias $[A \cdot Z \cdot A^T]$

$$\begin{array}{cccccccccc} a_1 & -\frac{1}{R_9} & 0 & -\frac{1}{R_0} & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{R_9} & a_2 & -\frac{1}{R_{10}} & 0 & -\frac{1}{R_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{R_{10}} & a_3 & 0 & 0 & -\frac{1}{R_2} & 0 & 0 & 0 & 0 \\ -\frac{1}{R_0} & 0 & 0 & a_4 & -\frac{1}{R_{11}} & 0 & -\frac{1}{R_3} & 0 & 0 & 0 \\ 0 & -\frac{1}{R_1} & 0 & -\frac{1}{R_{11}} & a_5 & -\frac{1}{R_{12}} & 0 & -\frac{1}{R_4} & 0 & 0 \\ 0 & 0 & -\frac{1}{R_2} & 0 & -\frac{1}{R_{12}} & a_6 & 0 & 0 & 0 & -\frac{1}{R_5} \\ 0 & 0 & 0 & -\frac{1}{R_3} & 0 & 0 & a_7 & -\frac{1}{R_{13}} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{R_4} & 0 & -\frac{1}{R_{13}} & a_8 & -\frac{1}{R_{14}} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{R_5} & 0 & -\frac{1}{R_{14}} & a_9 & 0 \end{array}$$

$$a_1 = \frac{1}{R_9} + \frac{1}{R_0}$$

$$a_2 = \frac{1}{R_{10}} + \frac{1}{R_9} + \frac{1}{R_0}$$

$$a_3 = \frac{1}{R_2} + \frac{1}{R_{10}}$$

$$\begin{aligned}
 a_4 &= \frac{1}{R_3} + \frac{1}{R_{11}} + \frac{1}{R_{10}} \\
 a_5 &= \frac{1}{R_4} + \frac{1}{R_{12}} + \frac{1}{R_{11}} + \frac{1}{R_1} \\
 a_6 &= \frac{1}{R_5} + \frac{1}{R_2} + \frac{1}{R_{12}} \\
 a_7 &= \frac{1}{R_{13}} + \frac{1}{R_3} + \frac{1}{R_6} \\
 a_8 &= \frac{1}{R_{13}} + \frac{1}{R_{14}} + \frac{1}{R_7} + \frac{1}{R_4} \\
 a_9 &= \frac{1}{R_5} + \frac{1}{R_{14}} + \frac{1}{R_8}
 \end{aligned}$$

La forma de construir la matriz de conductancias puede hacerse sin el uso de la matriz incidencia, pero ilustra la forma en que se posicionan los elementos en la matriz para formar el caso general en la computadora.

La forma general es

$$\begin{bmatrix}
 2 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
 -1 & 3 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
 0 & -1 & 3 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\
 0 & 0 & -1 & 3 & -1 & 0 & 0 & 0 & -1 & 0 \\
 0 & 0 & 0 & -1 & 2 & 0 & 0 & 0 & 0 & -1 \\
 -1 & 0 & 0 & 0 & 0 & 3 & -1 & 0 & 0 & 0 \\
 0 & -1 & 0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 \\
 0 & 0 & -1 & 0 & 0 & 0 & -1 & 4 & -1 & 0 \\
 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 4 & -1 \\
 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 3
 \end{bmatrix}$$

con número de condicinamiento: $(4 + \sqrt{5}) \left(\frac{3}{2} + \frac{1}{2}\sqrt{5}\right) : 16.326$

que como en el caso cilíndrico indica que esta bien condicionada y puede ser trabajada con precisión simple, aunque de cierta manera indique que es un poco menos estable que la cilíndrica.

$$-A \cdot i_b = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

2.8 Análisis de error en una red eléctrica

En general el análisis de error es el estudio y evaluación de las incertidumbres en una medida, al medir una o más cantidades x, y, \dots , con sus correspondientes incertidumbres $\delta x, \delta y, \dots$, y queremos utilizar estas cantidades medidas para encontrar una cantidad real de interés q , entonces el problema principal se concentra en saber cómo estas incertidumbres $\delta x, \delta y, \dots$, se propagan a través de los cálculos para obtener el correspondiente δq .

El conocer cómo se ve afectado el error δq por los errores individuales de sus elementos permitiría tener un control de este error final. Como se conoce el error o incertidumbre de los elementos individuales y también el valor exacto de estos elementos, se puede evaluar el error final, provocado por los errores individuales; esto es, que la incertidumbre en una función de varias variables queda:

Supóngase que x, \dots, z son medidas con incertidumbres $\delta x, \dots, \delta z$ y estos valores medidos son usados para calcular la función $q(x, \dots, z)$. Si las incertidumbres en x, \dots, z , son independientes y aleatorias entonces la incertidumbre en q es

$$\delta q = \sqrt{\left(\frac{\partial q}{\partial x} \delta x\right)^2 + \dots + \left(\frac{\partial q}{\partial z} \delta z\right)^2}$$

que nunca es más grande que la suma ordinaria

$$\delta q \leq \left| \frac{\partial q}{\partial x} \right| \delta x + \dots + \left| \frac{\partial q}{\partial z} \right| \delta z.$$

Para nuestro problema la función de varias variables es el potencial en la red eléctrica que depende de los elementos individuales o resistencias, en el circuito se conocen los valores nominales de los resistores así como sus tolerancias o incertidumbres. Aquí se tiene, como se vio en la analogía de resistencias eléctricas, el medio es modelado pero se idealiza que los valores así obtenidos están exentos de error y es claro que en la realidad esto no es así.

Las resistencias geoelectricas construidas, edifican al modelo en su totalidad, y es claro que cuando la geometría de la malla es coincidente con la geometría del modelo, las resistencias eléctricas no incluyen ningún error en su cálculo pues se crean geoméricamente perfectas, esto es, no hay error en ninguno de los parámetros de su creación, como son que se conoce la resistividad entre nodos, la coincidencia del vector densidad de corriente con la malla y las dimensiones de las celdas.

En el caso de la malla esférica y el medio ideal horizontal, se tiene el problema que no se conocen los valores nominales de algunos elementos pero el error en cada uno de ellos si se puede controlar:

El error en una celda puede ser visto como en la figura 2.8.

En un circuito eléctrico se denota cómo v_0 la función de salida que depende de los valores de la resistencia y la corriente:

$$v_0 = v_0(R_1, R_2, \dots, R_n; i_i)$$

y el error

$$\delta v_0 = \sqrt{\left(\frac{\partial v_0}{\partial R_1} \delta R_1 \right)^2 + \left(\frac{\partial v_0}{\partial R_2} \delta R_2 \right)^2 + \dots + \left(\frac{\partial v_0}{\partial R_n} \delta R_n \right)^2}$$

Donde $\delta R_1, \delta R_2, \dots, \delta R_n$ son los errores en las resistencias individuales.

Al conocer la topología del circuito es posible construir la matriz incidencia asociada y hacer un análisis de error. Como ejemplo se aplicará el análisis al circuito serie paralelo de la figura 2.9:

Con matriz incidencia

$$\begin{bmatrix} 1 & 1 & 0 & -1 \\ -1 & -1 & 1 & 0 \end{bmatrix}$$

y con función de salida

$$v_0 = \frac{R_1 R_2}{R_1 + R_2} + R_3$$

La función error

$$\delta v_0^2 = \delta R_3^2 + \delta R_2^2 \left(- \left(\frac{R_1 R_2}{(R_1 + R_2)^2} \right) + \frac{R_1}{R_1 + R_2} \right)^2 + \delta R_1^2 \left(- \left(\frac{R_1 R_2}{(R_1 + R_2)^2} \right) + \frac{R_2}{R_1 + R_2} \right)^2$$

Si los errores en las resistencias R_2 y R_3 son cero, asumimos que están perfectamente calculadas y que además tienen los mismos valores nominales, entonces se puede observar la función error dependiente únicamente de los valores que toma R_1 y como su variación hace la función error máxima o mínima.

Siendo $R_2 = R_3 = 1 \Omega$, y $\delta R_1 = 0.1 R_1$

$$\delta v_0(R_1) = 0.1 R_1 \left(\frac{1}{R_1 + 1} \right)^2$$

En la figura 2.10 se observa, como el error o tolerancia máxima de la resistencia final se tiene, cuando la resistencia R_1 es igual al valor de las resistencias R_2 y R_3 , la resistencia total tiene una menor tolerancia o rango de error si

$$R_1 \ll (R_2; R_3)$$

ó si

$$R_1 \gg (R_2; R_3)$$

Para el caso de que las resistencias R_1 y R_2 sean iguales en valor nominal y se analice el error, debido a las variaciones de la resistencia R_3 , con una tolerancia en ella de $0.1 R_3$ y la figura 2.11.

El error es mayor cuando la resistencia R_3 es mucho mayor que los valores nominales de R_1 y R_2 . La corriente circula casi en su totalidad por la resistencia R_3 y su efecto en la resistencia final se incrementa, mientras que cuando R_3 es menor, la distribución del error se hace más equitativa y dependiente de las tres resistencias y su tolerancia.

Es obvio que la resistencia es más exacta entre más pequeña sea respecto a las demás pues su efecto disminuye, así como también cuando es mucho más grande que los otros elementos, la tolerancia de la resistencia analizada influye en los resultados y es mucho más fácil obtener una resistencia con una tolerancia grande que con una tolerancia fina, una resistencia con un error grande es el caso de la malla esférica y las fronteras no coincidentes con ella, la resistencia no es conocida exactamente en la interfase de dos medios, pero si conocemos la resistencia exacta de la mayoría de las resistencias y si incluimos un rango de tolerancia en las resistencias de la interface, podríamos saber qué resistencias modificar para disminuir el error en la salida. Mientras que en la malla el error se propaga con la distancia

2.9 Sensibilidad en una red eléctrica

La sensibilidad en un circuito es una parte importante en el diseño del modelo resistivo, la teoría de circuitos ha desarrollado la sensibilidad para determinar cómo varía la salida de un circuito respecto a cambios en sus elementos individuales, en el apartado anterior, se vio que el error o tolerancia de la salida podía ser analizado por los errores individuales y esto era con la ayuda de la función error, en el caso de la sensibilidad se utilizan las propiedades topológicas para encontrar los rangos de la salida, dados los rangos de sus resistencias, lo que se pretende es encontrar para qué rangos la red no trabajaría ó cual es el peor caso.

Para determinar el peor caso se debe encontrar la relación entre la variación de la salida y la variación o perturbación de cada elemento.

Si se denota v_0 el voltaje de salida de la red \mathcal{N} , esta compuesta por los resistores R_1, R_2, \dots, R_n

y supóngase que la corriente de entrada es i_i . Entonces v_0 es función de la

corriente de entrada y de los valores de sus resistencias: $v_0 = v_0(R_1, R_2, \dots, R_n; i_i)$

Dados $R_1^o, R_2^o, \dots, R_n^o$, representando a los valores nominales y v_0^o la correspondiente salida nominal:

$$v_0^o = v_0(R_1^o, R_2^o, \dots, R_n^o; i_i)$$

Formalmente la *sensibilidad* de \aleph con respecto a cualquier valor de resistencia R_k es la derivada parcial de $v_0(R_1, R_2, \dots, R_n; i_i)$ con respecto a R_k evaluado en los valores resistivos nominales $R_1^o, R_2^o, \dots, R_n^o$, multiplicado por (R_k/v_0) :

$$S_{R_k}^{v_0} = \left(\frac{R_k}{v_0} \right) \frac{\partial v_0(R_1, R_2, \dots, R_n; i_i)}{\partial R_k} \Bigg|_{\substack{R_1 = R_1^o \\ R_2 = R_2^o \\ \cdot \\ \cdot \\ R_n = R_n^o \\ i_i = 1}}$$

que también es conocida como sensibilidad normalizada, mientras que la sensibilidad sin normalizar:

$$\hat{S}_{R_k}^{v_0} = \frac{\partial v_0(R_1, R_2, \dots, R_n; i_i)}{\partial R_k} \Bigg|_{\substack{R_1 = R_1^o \\ R_2 = R_2^o \\ \cdot \\ \cdot \\ R_n = R_n^o \\ i_i = 1}}$$

Para el problema analizado en análisis de error

$$v_0 = \left(\frac{R_1 R_2}{R_1 + R_2} + R_3 \right) i_i$$

siendo $R_1 = 8\Omega, R_2 = R_3 = 1\Omega$ ($R_1 \ll (R_2, R_3)$)

$v_0 = 1.88 \text{ volts}$. La sensibilidad sin normalizar de v_0 con respecto a R_1 :

$$\hat{S}_{R_1}^{v_0} = \left. \frac{\partial v_0}{\partial R_1} \right|_{\substack{R_1=8 \\ R_2=1 \\ R_3=1 \\ i_i=1}} = - \left(\frac{R_1 R_2}{(R_1 + R_2)^2} \right) + \frac{R_2}{R_1 + R_2} \Bigg|_{\substack{R_1=8 \\ R_2=1 \\ R_3=1 \\ i_i=1}} = 0.012$$

La sensibilidad normalizada es:

$$S_{R_1}^{v_0} = \frac{R_1}{v_0} \hat{S}_{R_1}^{v_0} = \left(\frac{8}{1.88} \right) 0.012 = 0.051$$

Considérese un cambio de 10 porciento en el valor de R_1 , esto es $R_1 = 8.8$

$$v_0(R_1, R_2^c, R_3^c; i_i) = R_1 \frac{\hat{S}_{R_1}^{v_0}}{S_{R_1}^{v_0}} = 8.8 \left(\frac{0.012}{0.051} \right) = 2.07$$

Que indica cómo un incremento en la resistencia nominal del 10 porciento implica un incremento en la salida. También indica que al ser la sensibilidad positiva un decremento pequeño en el valor nominal producirá un decremento en la salida. Para un decremento de 10 porciento en R_1

$$v_0(R_1, R_2^c, R_3^c; i_i) = 7.2 \left(\frac{0.012}{0.051} \right) = 1.694$$

Ahora véase la sensibilidad de R_2 , siendo la sensibilidad sin normalizar de v_0 con respecto a R_2 :

$$\hat{S}_{R_2}^{v_0} = \left. \frac{\partial v_0}{\partial R_2} \right|_{\substack{R_1=8 \\ R_2=1 \\ R_3=1 \\ i_i=1}} = - \left(\frac{R_1 R_2}{(R_1 + R_2)^2} \right) + \frac{R_1}{R_1 + R_2} \Bigg|_{\substack{R_1=8 \\ R_2=1 \\ R_3=1 \\ i_i=1}} = 0.790$$

La sensibilidad normalizada es:

$$S_{R_2}^{v_0} = \frac{R_2}{v_0} \hat{S}_{R_2}^{v_0} = \left(\frac{1}{1.88} \right) 0.790 = 0.420$$

Considérese un cambio de 10 porciento en el valor de R_2 , esto es $R_2 = 1.1$

$$v_0(R_2, R_1^c, R_3^c; i_i) = R_2 \frac{\hat{S}_{R_2}^{v_0}}{S_{R_2}^{v_0}} = 1.1 \left(\frac{0.790}{0.420} \right) = 2.069$$

Al estar las dos resistencias en paralelo, su sensibilidad positiva muestra que un incremento en sus valores incrementa la salida, pero dado que la resistencia en R_1 es mucho mayor, la salida es mas sensible que en el caso de R_2

Para R_3 :

$$\hat{S}_{R_3}^{v_0} = \left. \frac{\partial v_0}{\partial R_3} \right|_{\substack{R_1=8 \\ R_2=1 \\ R_3=1 \\ i_i=1}} = 1$$

La sensibilidad normalizada:

$$S_{R_3}^{v_0} = \frac{R_3}{v_0} \hat{S}_{R_3}^{v_0} = 0.531$$

La sensibilidad es un poco mayor que en el caso de R_2 , pero al seguir siendo positiva, indica como las variaciones de todos los elementos son proporcionales a las variaciones de la salida, esto concuerda con el análisis de error, hecho anteriormente en donde la tolerancia final es mínima al incrementar la resistencia R_1 .

En general, si la sensibilidad de la salida con respecto a una resistencia fue positiva, la salida se incrementará (decrementará) si el valor de la resistencia se incrementa (decrementa), de forma similar si la sensibilidad de la salida con respecto a un resistor fue negativa, la salida decrecerá (incrementará) si el valor de la resistencia se incrementa (decrece).

2.10 La red adjunta como aproximación de sensibilidad

Las leyes de Kirchhoff de corriente y voltaje son construcciones algebraicas que resultan de la interconexión de los elementos en una red y son independientes de las características de los elementos, ellas implican la conservación de la energía y no necesitan ser sumadas a los postulados de la teoría de redes, entonces es posible generalizar resultados.

Considerando una red $\hat{\mathcal{N}}$ con la misma configuración topológica, los mismos voltajes y corrientes de referencia y el mismo número de lazos como en una primer red \mathcal{N} para la que se cumple:

$$A\mathbf{i} = \mathbf{0}$$

$$\mathbf{v} = A^T \mathbf{v}_n$$

Como resultado, ambas redes tienen la misma matriz incidencia y las ecuaciones que se cumplen para la primera son aplicadas a la segunda:

$$A\hat{\mathbf{i}} = \mathbf{0}$$

$$\hat{\mathbf{v}} = A^T \hat{\mathbf{v}}_n$$

donde $\hat{\mathbf{v}}_n$ es el vector de voltaje nodal de la red $\hat{\mathcal{N}}$ llamada la red adjunta y se cumple

$$\sum_{K=1}^b v_k \hat{i}_k = \mathbf{v}^T \hat{\mathbf{i}} = (A^T \mathbf{v}_n)^T \hat{\mathbf{i}} = \mathbf{v}_n^T (A\hat{\mathbf{i}}) = \hat{\mathbf{v}}_n^T (\mathbf{0}) = 0$$

y una derivación similar da:

$$\sum_{K=1}^b \hat{v}_k i_k = \hat{\mathbf{v}}^T \mathbf{i} = 0.$$

Conocidas dichas relaciones como el teorema de Tellegen.

Aplicando el teorema a ambas redes, con su correspondiente perturbación en corrientes y voltajes y siendo la relación del k -ésimo ($k=1, \dots, n$) lazo de la red \mathcal{N} :

$$v_k = R_k i_k$$

Si se perturba cada resistor, entonces los voltajes y corrientes de lazo se perturbarán también. Si ΔR_k denota el cambio en una resistencia, Δv_k el cambio en voltajes, y Δi_k el cambio en corrientes del k -ésimo lazo, entonces:

$$(v_k + \Delta v_k) = (R_k + \Delta R_k)(i_k + \Delta i_k)$$

para $k=1, 2, \dots, n$.

Expandiendo:

$$v_k + \Delta v_k = R_k i_k + \Delta R_k i_k + R_k \Delta i_k + \Delta i_k \Delta R_k$$

como $v_k = R_k i_k$ y $\Delta i_k \Delta R_k$ es muy pequeño

$$\Delta v_k \simeq \Delta R_k i_k + R_k \Delta i_k$$

En el teorema de Tellegen para la red \hat{N} , siendo \hat{i}_k , la corriente del k-ésimo lazo.

$$\sum_{k=1}^{n_b} (v_k + \Delta v_k) \hat{i}_k = 0$$

y

$$\sum_{k=1}^{n_b} (i_k + \Delta i_k) \hat{v}_k = 0$$

combinándolas se obtiene:

$$\sum_{k=1}^{n_b} (\Delta v_k \hat{i}_k - \Delta i_k \hat{v}_k) = 0$$

ó

$$\sum_{k=1}^{n_b} ((\Delta R_k i_k + R_k \Delta i_k) \hat{i}_k - \Delta i_k \hat{v}_k) = 0$$

$$\sum_{k=1}^{n_b} [(R_k \hat{i}_k - \hat{v}_k) \Delta i_k + (i_k \hat{i}_k) \Delta R_k] = 0$$

Pero también se cumple

$$\Delta v_i \hat{i}_i + \Delta v_0 \hat{i}_0 \sum_{k=1}^{n_b} [(R_k \hat{i}_k - \hat{v}_k) \Delta i_k + (i_k \hat{i}_k) \Delta R_k] = 0$$

Si en la red adjunta $\hat{i}_i = 0, \hat{i}_0 = 1, R_k \hat{i}_k - \hat{v}_k = 0$ y se analiza para la resistencia R_l

queda

$$\Delta v_0 = -(i_l \hat{i}_l) \Delta R_l$$

ó

$$\hat{S}_{R_l}^{v_0} = \frac{\Delta v_0}{\Delta R_l} = -i_l \hat{i}_l$$

se obtiene que la sensibilidad es el producto de las corrientes de ambas redes, las derivadas parciales son sustituidas con sólo este producto, pero hay que realizar el

análisis de las dos redes primero, la diferencia en la matriz adjunta la hace la fuente de corriente pues en la adjunta se hace cero y la que antes era electrodo de medición se transforma en electrodo de corriente para la adjunta.

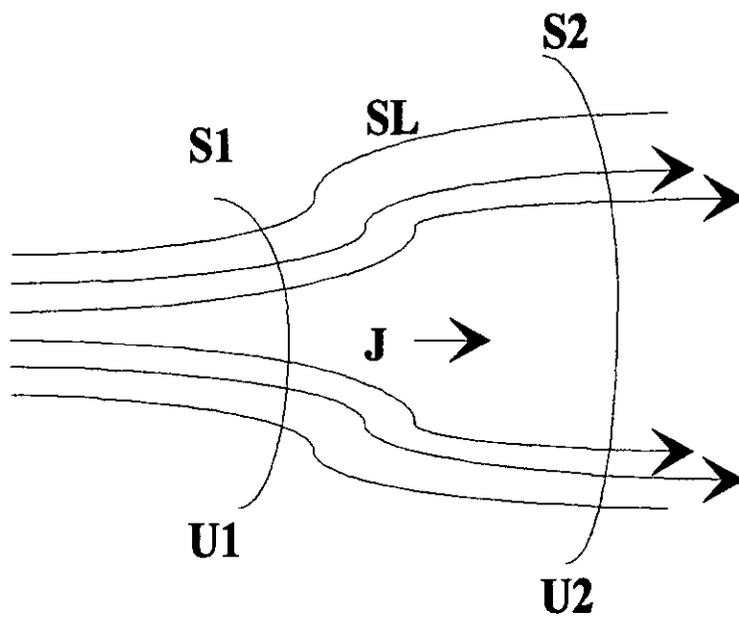


Figura 2.1: Tubo de Corriente

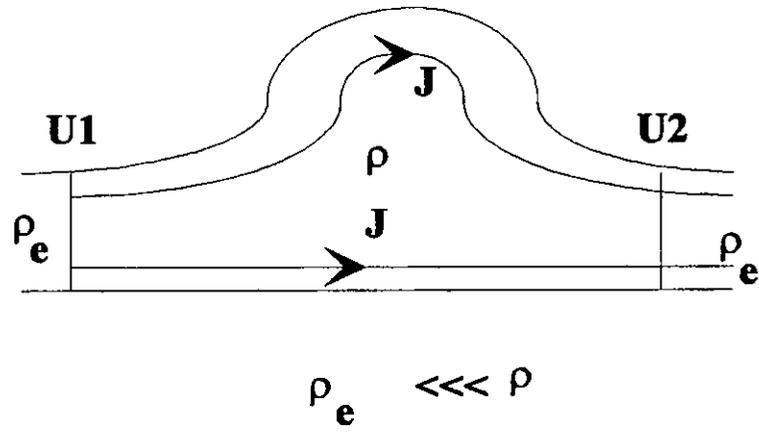


Figura 2.2: Lineas de corriente

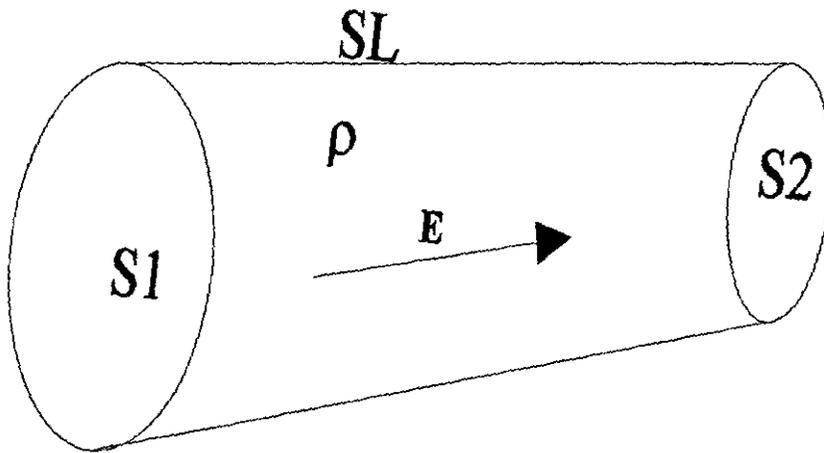


Figura 2.3: Cilindro homogéneo

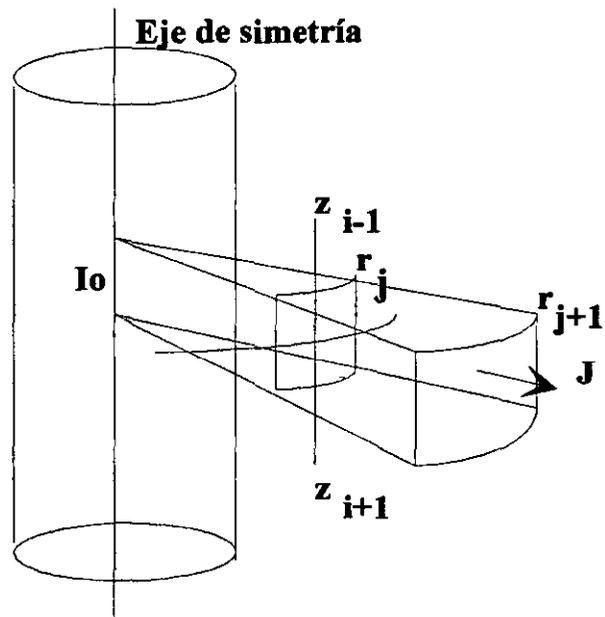


Figura 2.4: Sistema Cilíndrico

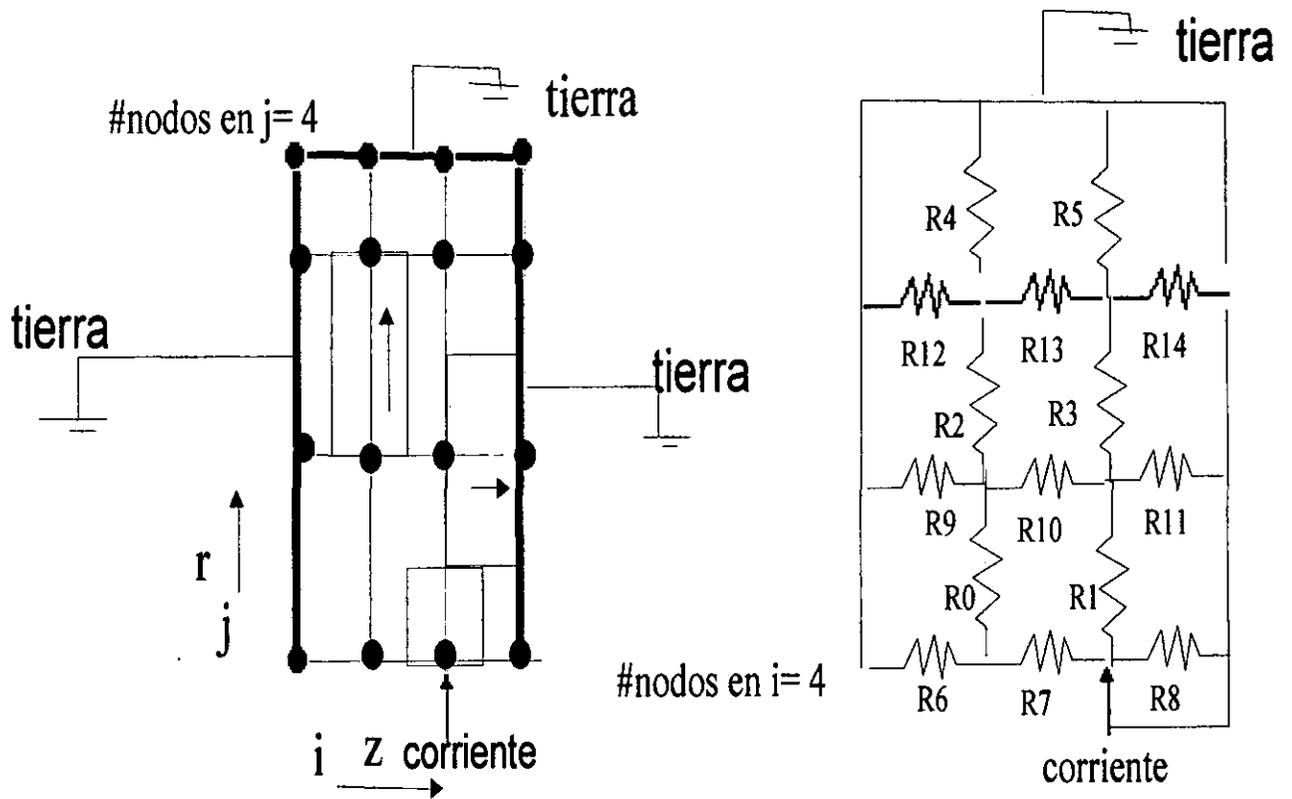


Figura 2.5: Circuito cilíndrico

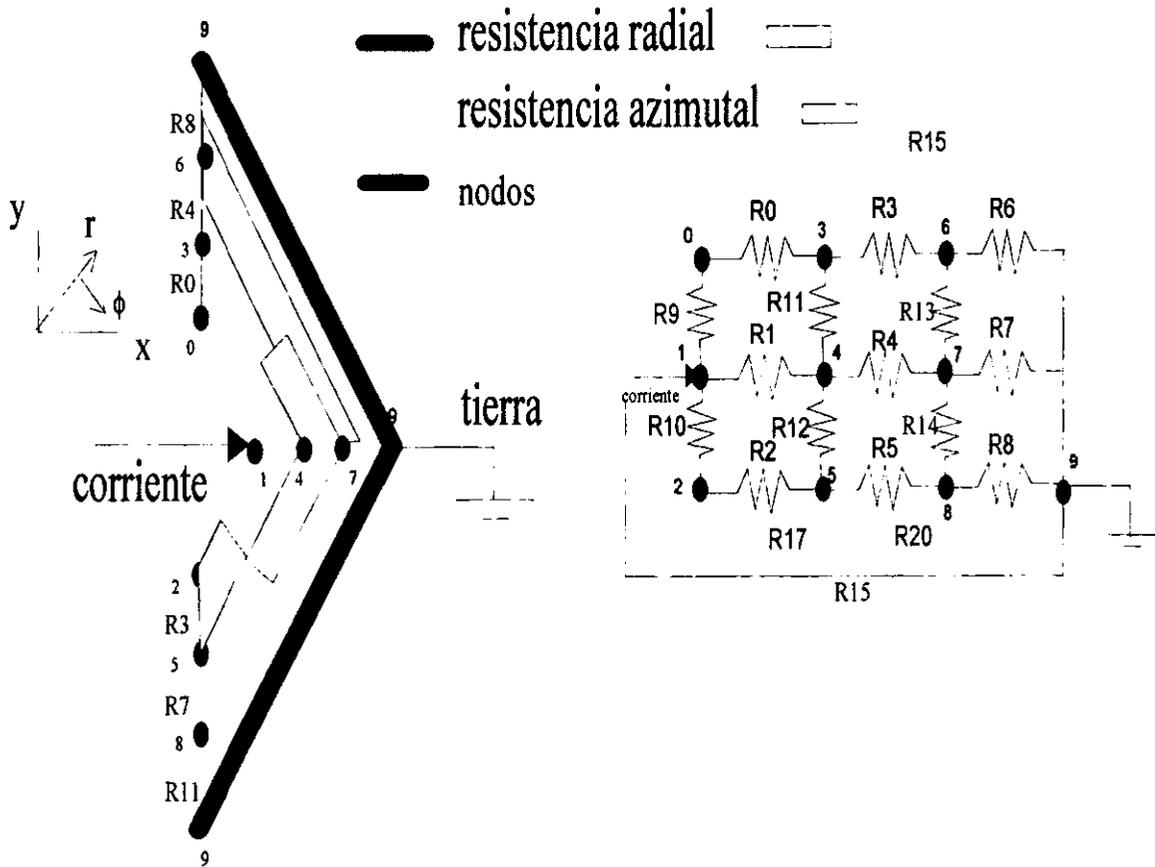


Figura 2.6: Circuito esférico

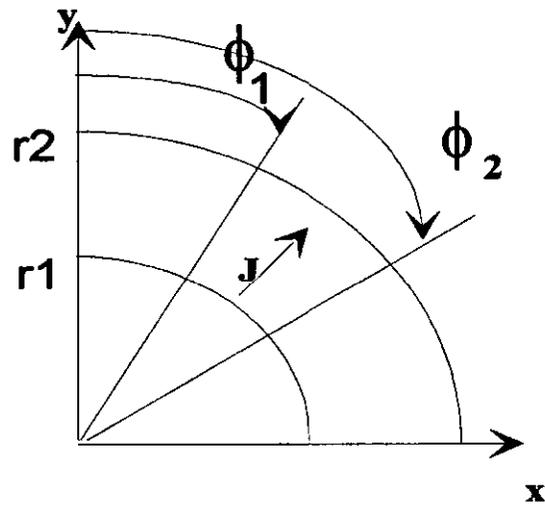


Figura 2.7: Sistema esférico

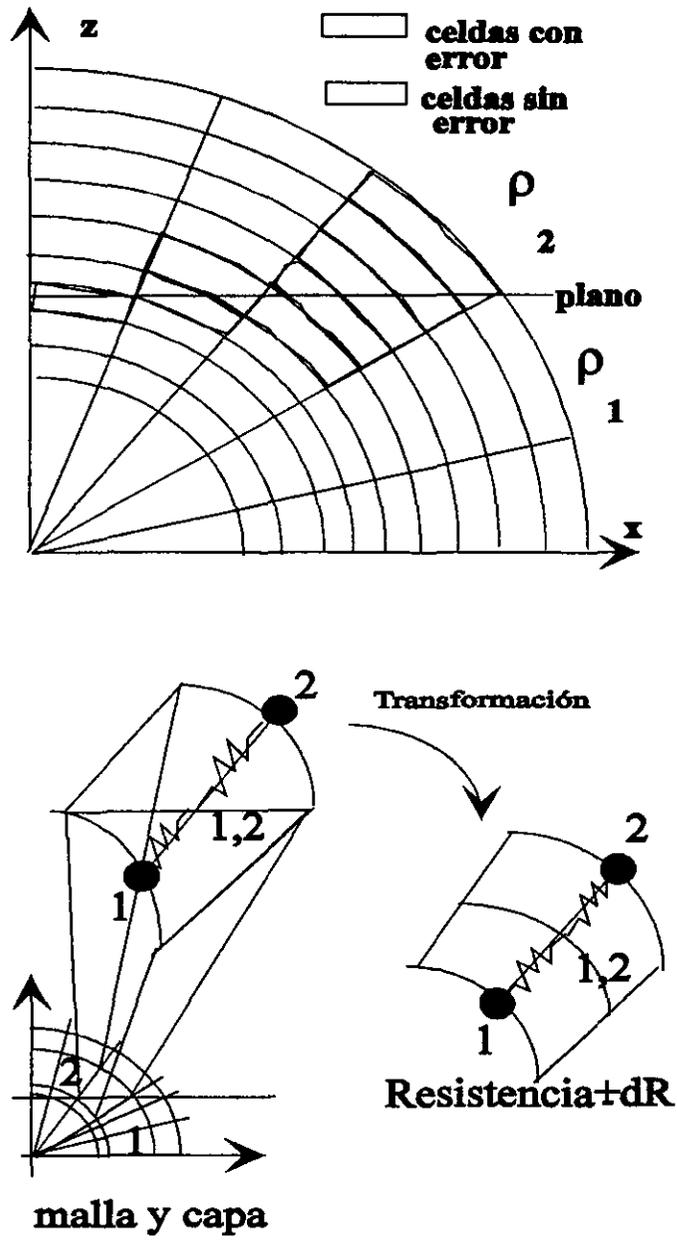


Figura 2.8: Error en una celda

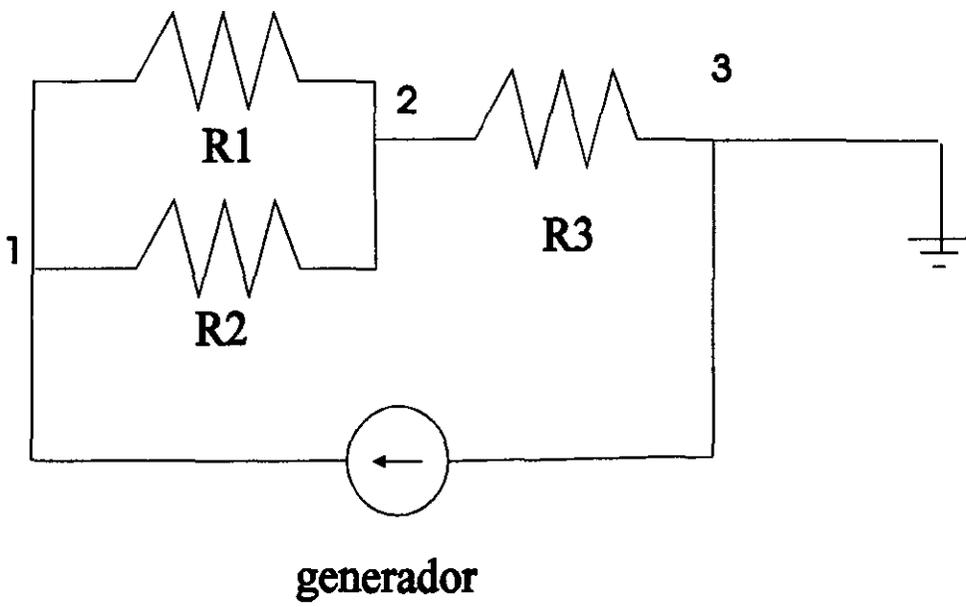


Figura 2.9: Circuito

Función Error de R1

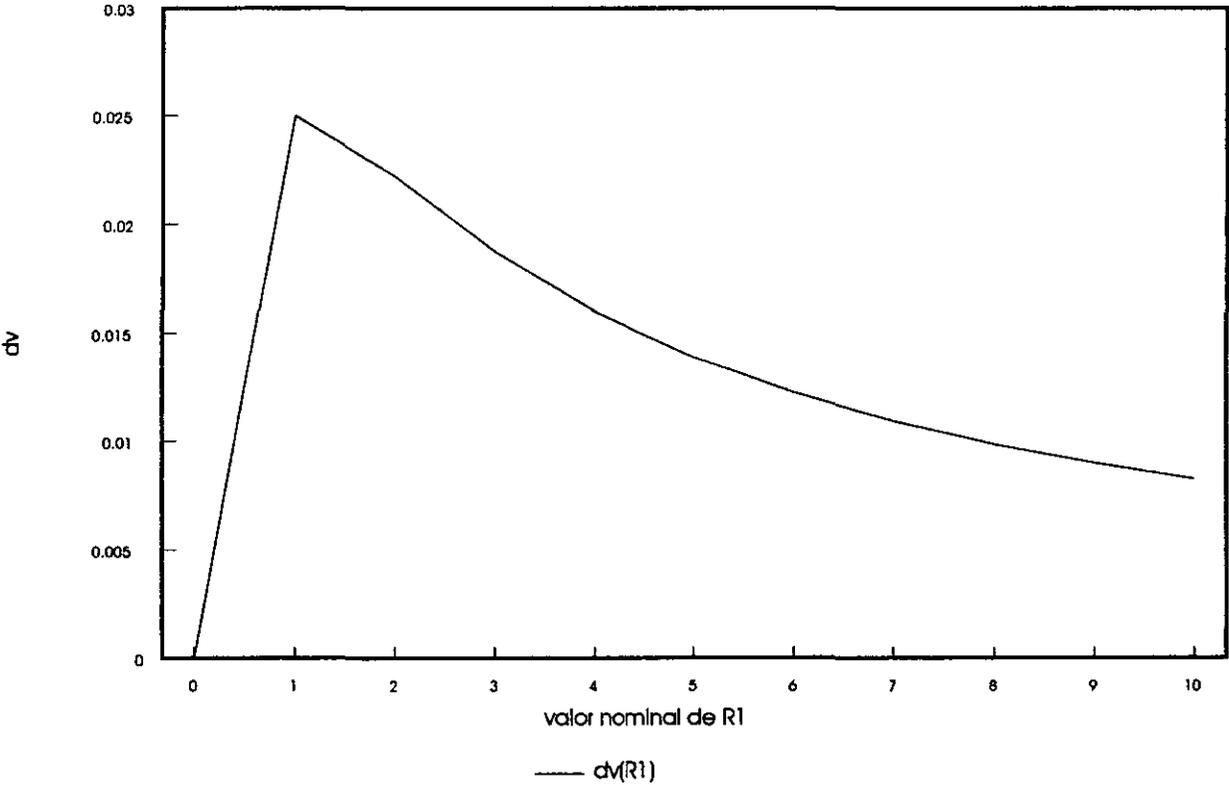


Figura 2.10: Función error de R1

Función Error de R3

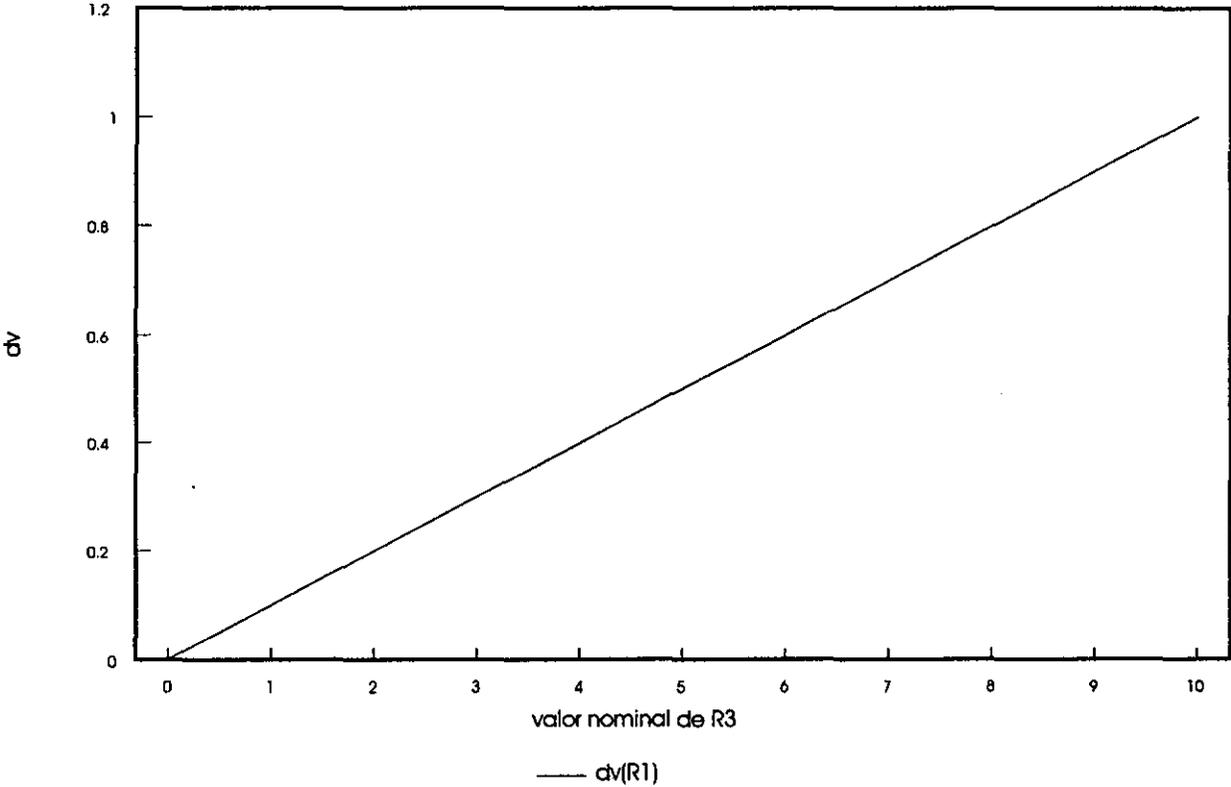


Figura 2.11: Función error de R3

Capítulo 3

Aplicación de la Teoría

3.1 Introducción

En los anteriores capítulos se obtuvieron las herramientas para lograr el modelado. Los algoritmos en diferencias finitas y resistencias eléctricas, fueron desarrollados para el medio homogéneo y una sola fuente de corriente en el caso de la red eléctrica o una fuente puntual de corriente en diferencias finitas, para ambos métodos se tiene el caso esférico y el caso cilíndrico.

Para la malla esférica con resistencias, se discretizó el medio de acuerdo a la geometría del problema, se observaron los resultados con dos diferentes topologías, en las dos el modelo esférico se realizó con una Dona(esfera homogénea con una cavidad esférica en su centro) de un radio interno y otro externo, aunque ambos modelos esféricos consideran corrientes azimutales y radiales la localización de la fuente en los dos es diferente.

En el primer modelo que se estudió utilizando la matriz incidencia no se consideran corrientes azimutales en el radio interno y en el radio externo, de tal forma que las resistencias inmediatas a la fuente únicamente están en paralelo, esto origina que la matriz de coeficientes NO SEA PENTADIAGONAL, pues la fuente tiene relación al mismo tiempo con las resistencias en paralelo.

El objetivo entonces, era determinar que radio externo e interno debía escogerse así como la discretización en forma angular y radial para obtener una buena

aproximación al valor teórico del potencial, se observó la distancia de 0.4065m que es el radio de observación de la normal corta teórica, se tienen como resultados en medio homogéneo con este modelo lo siguiente:

a) El número de nodos en la dirección radial no influye en el valor de observación.

b) En el medio homogéneo el número de nodos en dirección azimutal no influye, pues es homogéneo y se verificó que no influyen en la dirección azimutal las corrientes.

c) El parámetro que controla el error en los resultados es el radio externo, mientras que el radio interno puede alcanzar valores tan pequeños como sea posible, pero no influye en los resultados.

Modelo 2

Este modelo cambia la topología del circuito en el radio interno, aquí, sí se tienen resistencias azimutales, y la fuente de corriente puede ser localizada en cualquiera de los nodos del radio interno.

a) No presenta dependencia del número de nodos en la dirección r y azimutal, además sigue siendo muy estable como en el caso anterior.

b) La localización de la fuente en el origen de coordenadas con este modelo sí influye en los resultados del potencial.

c) El parámetro que controla el error en los resultados es el radio externo.

La malla cilíndrica con resistencias

Aquí las condiciones de la malla mostraron resultados diferentes:

a) Se debe encontrar el número de nodos y factor de expansión adecuados para obtener resultados, difiere mucho del caso esférico de resistencias.

b) Para obtener un error pequeño debe ser la distancia radial de apertura de la malla mayor que la longitud axial del cilindro, si no se tiene factor de expansión en la dirección z .

c) Se presenta dependencia del radio interno así como de la longitud del cilindro.

El paso siguiente es introducir las mallas al medio con contrastes de resistividad, el proceso es como sigue para ambas geometrías:

1. Dada la profundidad del electrodo de corriente y conociendo las dimensiones de la malla, se le proporciona información sobre las capas para que se rellenen los respectivos nodos con los respectivos contrastes.
2. Se formanda la matriz de coeficientes.
3. Se resuelve el sistema.
4. Se encuentra la resistividad aparente para el nodo del electrodo de potencial.

3.2 Contrastes de resistividad

Tanto en diferencias finitas como en red eléctrica se presenta el problema de como evaluar la resistividad entre nodos, en el caso homogéneo esto no es problema, pero en el caso de contrastes resistivos si.

Suponiendo que se tiene un medio de conductividad σ_1 y otro de conductividad σ_2 . Se quiere obtener el valor intermedio σ_{12} , entonces se colocan dos nodos, uno en cada medio y separados a la misma distancia de la frontera, como sería el caso del modelo cilíndrico y capas horizontales, si suponemos que la interpolación que resuelve el problema es solamente:

$$\sigma_{12} = \frac{\sigma_1 + \sigma_2}{2}$$

Entonces, debe funcionar para casos extremos como $\sigma_1 = 0$ y $\sigma_2 = 100$, o sea un medio que no conduce con otro que si lo es se tendría: $\sigma_{12} = 50$, que significaría que en la frontera se conduce la mitad y no es así.

Si se hacen analogías con resistencias, entonces se colocan entre los nodos a-ab la resistencia R_1 de conductividad σ_1 y ab-a la resistencia R_2 de conductividad σ_2 , la suma en serie queda:

$$R = R_1 + R_2$$

y de la definición de resistencia

$$R = \frac{1}{\sigma} \frac{L}{S}$$

L=longitud

$S = \text{área}$

Además si las dos resistencias tienen la misma área S de dimensión unitaria:

$$R = \frac{1}{\sigma} (L_2 - L_1)$$

ó para las dos resistencias

$$R_1 = \frac{L_{ab} - L_a}{\sigma_1}; R_2 = \frac{L_b - L_{ab}}{\sigma_2}$$

entonces la suma en serie será:

$$\frac{L_b - L_a}{\sigma_{12}} = \frac{L_{ab} - L_a}{\sigma_1} + \frac{L_b - L_{ab}}{\sigma_2}$$

pero

$$L_{ab} = \frac{L_a + L_b}{2}$$

$$\frac{L_b - L_a}{\sigma_{12}} = \frac{\frac{L_a + L_b}{2} - L_a}{\sigma_1} + \frac{L_b - \frac{L_a + L_b}{2}}{\sigma_2}$$

$$\frac{L_b - L_a}{\sigma_{12}} = \frac{L_b - L_a}{2\sigma_1} + \frac{L_b - L_a}{2\sigma_2}$$

Finalmente

$$\sigma_{12} = \frac{2\sigma_1\sigma_2}{\sigma_1 + \sigma_2}$$

que aplicado al mismo problema extremo da $\sigma_{12} = 0$, que es un resultado más creíble y funciona cuando el medio es el mismo.

Se aplicará la función de interpolación en los coeficientes de conductividad intermedios como:

$$\sigma_{i,j+1/2} = \frac{2\sigma_{i,j+1}\sigma_{i,j}}{\sigma_{i,j+1} + \sigma_{i,j}}$$

Entonces la resistencia internodal cuando hay contrastes de resistividad toma la forma:

Para el caso cilíndrico:

En la dirección R entre los nodos (i,j) y $(i,j+1)$:

$$R_{i,j-i,j+1} = \frac{1}{2\pi(z_{i+1} - z_i)} \left(\frac{\sigma_{i,j+1} + \sigma_{i,j}}{2\sigma_{i,j+1}\sigma_{i,j}} \right) \ln \frac{r_{j+1}}{r_j}$$

En la dirección Z entre los nodos (i,j) y (i+1,j):

$$R_{i,j-i+1,j} = \frac{(z_{i+1} - z_i)}{\pi(r_{j+1}^2 - r_j^2)} \left(\frac{\sigma_{i+1,j} + \sigma_{i,j}}{2\sigma_{i+1,j}\sigma_{i,j}} \right)$$

que facilmente se ve que son análogas a las desarrolladas por Dakhnov:

En la dirección R:

$$R_{i,j-i,j+1} = \frac{1}{2\pi(z_{i+1} - z_i)} \left(\rho_{i,j} \ln \frac{r_{j+1} + r_j}{2r_j} + \rho_{i,j+1} \ln \frac{2r_{j+1}}{r_{j+1} + r_j} \right)$$

en la dirección Z:

$$R_{i,j-i+1,j} = \frac{(z_{i+1} - z_i)}{\pi(r_{j+1}^2 - r_j^2)} (\rho_{i,j} + \rho_{i+1,j})$$

Para el sistema esférico:

En la dirección R

$$R_{i,j-i,j+1} = \frac{1}{2\pi(\cos \phi_i - \cos \phi_{i+1})} \left(\frac{r_{j+1} - r_j}{r_{j+1} + r_j} \right) \left(\frac{r_{j+1}\rho_{i,j} + r_j\rho_{i,j+1}}{r_{j+1}r_j} \right)$$

En la dirección ϕ

$$R_{i,j-i+1,j} = \frac{1}{2\pi(r_{j+1} - r_i)} \left(\rho_{i,j} \ln \frac{\tan(\phi_{i+1} + \phi_i)/4}{\tan \phi_i/2} + \rho_{i+1,j} \ln \frac{\tan \phi_i/2}{\tan(\phi_{i+1} + \phi_i)/4} \right)$$

3.3 El Factor de Resistencia Terminal (FRT)

Debido a que las fronteras de las mallas cilíndricas y esféricas ya sean diferencias finitas o resistencias eléctricas, están simulando una tierra o cero potencial a determinada distancia que es la amplitud total de la malla o radio de visión, es necesario corregir los valores de potencial de los nodos que anteceden al nodo de tierra, con

un factor de corrección conocido en la literatura como Factor de Resistencia Terminal llamado así por Kim.

Este factor de corrección lo que hace es normalizar el valor anterior al nodo de tierra a su correcto valor en un medio homogéneo y se obtiene de la siguiente manera:

Siendo el potencial de los últimos nodos de la malla $U(imax,j)$ y el valor de los nodos anteriores a esta frontera $U(imax-1,j)$, entonces:

$$U_{(imax,j)}^{malla} = 0$$

$$U_{(imax-1,j)}^{homo} = K * (U_{(imax-1,j)}^{malla})$$

Donde k es el FRT

Entonces si la resistencia terminal entre los nodos $(imax,j)$ y $(imax-1,j)$ es

$$R_{[(imax)-(imax-1)]}$$

Y de la relación potencial- resistencia

$$\frac{(U_{(imax,j)}^{malla} - U_{(imax-1,j)}^{malla})}{R_{[(imax)-(imax-1)]}^{malla}} = k \frac{(U_{(imax,j)}^{homo} - U_{(imax-1,j)}^{homo})}{R_{[(imax)-(imax-1)]}^{homo}}$$

quedando

$$R_{[(imax)-(imax-1)]}^{homo} K = R_{[(imax)-(imax-1)]}^{malla}$$

En la malla esférica para el medio homogéneo, podemos observar como el error esta en el inicio o radio interno, asi como en el radio externo, pues se obliga a obtener el cero o tierra segun la grafica 3.1.

Al disminuir el número de nodos en la dirección radial, el error en el origen disminuye figura 3.2.

La disminución del factor radial a 1.15, solo afecta a la distancia exterior de la malla esférica figura 3.3.

Para la malla esférica, un aumento en el factor de expansión radial significa un mayor error en el origen, pero se mejoran los resultados en el radio externo figura 3.4.

En la siguiente, se analiza el efecto de incrementar el radio interno a 0.1m , con malos resultados para el origen, el origen para la normal corta fue de 0.01m figura 3.5.

Como se mencionó antes, el radio externo indica la distancia a la que esta el cero o tierra, y esto lleva un error que puede corregirse para nodos anteriores al radio externo con el factor de resistencia terminal (FRT), pero existe la posibilidad de abrir más allá la malla y obtener buenos resultados como hasta el rango de la herramienta lateral figura 3.6.

Para mostrar el FRT se aplicó en una de las graficas anteriores, nótese que el factor solo se aplica en la parte exterior, y factor de expansión radial ayuda a disminuir el efecto en el origen causado por el radio interno figura 3.7.

3.4 Cilíndrica vs esférica en la normal corta

En la figura 3.8 se comparan la malla cilíndrica de diferencias finitas y la malla cilíndrica de resistencias eléctricas para un contraste de 2 ohm-m, 50 ohm-m y 2 ohm-m, para un espesor igual en las tres de 10 m; como se puede ver las dos gráficas presentan la misma forma en sus respectivos errores absolutos, se presentan errores entre 1% y 10% en las fronteras, al final de la capa intermedia en 1019.5 se tiene un error del 29%, debido a que se esta utilizando un número de nodos en la dirección vertical de 31, es necesario incrementar el numero de nodos para reducir el error, este mismo efecto se presenta en el metro 1019.5, esto quiere decir que se esta teniendo un error mayor a la entrada de la capa resistiva, pues es el lado en donde se hace la medición y en el algoritmo se introduce el nodo de medición de la normal corta como puede verse en la figura 3.9 . Por otra parte se puede ver que no hay mucha diferencia entre diferencias finitas y las resistencias eléctricas en la geometría cilíndrica.

En la figura 3.10se tiene el mismo contraste de resistividad y espesores para

la malla esférica en resistencias eléctricas y diferencias finitas, se nota que los errores no son tan parecidos como en la geometría cilíndrica para las diferencias finitas y resistencias eléctricas, también hay errores mayores de 10% antes de entrar a la capa, pero se aprecia que dentro de la capa intermedia se tienen errores de menos de 1%.

En la figura 3.11 incremento la resistividad de la capa intermedia de 50 a 500 ohm-m, sin modificar los espesores para la geometría cilíndrica y esférica, aquí el rango de error es muy parecido al contraste inicial, como se ve la mayoría de los puntos de la geometría cilíndrica y esférica presentan errores entre 1% y 10%, aunque parece ser que en la esfera tienden a distribuirse mejor hacia 1% y además se tiene una reducción de nodos significativa.

En la figura 3.12 se presenta un contraste diferente de 500 ohm-m, 2 ohm-m y 500 ohm-m, para el mismo espesor que en los otros ejemplos. En la esférica fue necesario incrementar el número de nodos de 14 a 20 en la dirección angular, la radial se mantuvo fija en 20, los errores en la frontera siguen existiendo para las dos geometrías, aunque en el cilindro parece que también se incrementa el error, los contrastes grandes presentan problemas para los dos algoritmos.

Por último se tiene una comparación entre la malla cilíndrica y esférica para una sola profundidad figura 3.13, esto es como se ve el potencial para una sola posición, el electrodo de corriente está a 1m de la frontera, está dentro de la capa resistiva de 50 ohm-m. La malla esférica muestra como a 1m de distancia de la frontera, las equipotenciales parecen no distorsionarse mucho, en comparación a la malla cilíndrica en donde en las mismas condiciones la deformación es mayor.

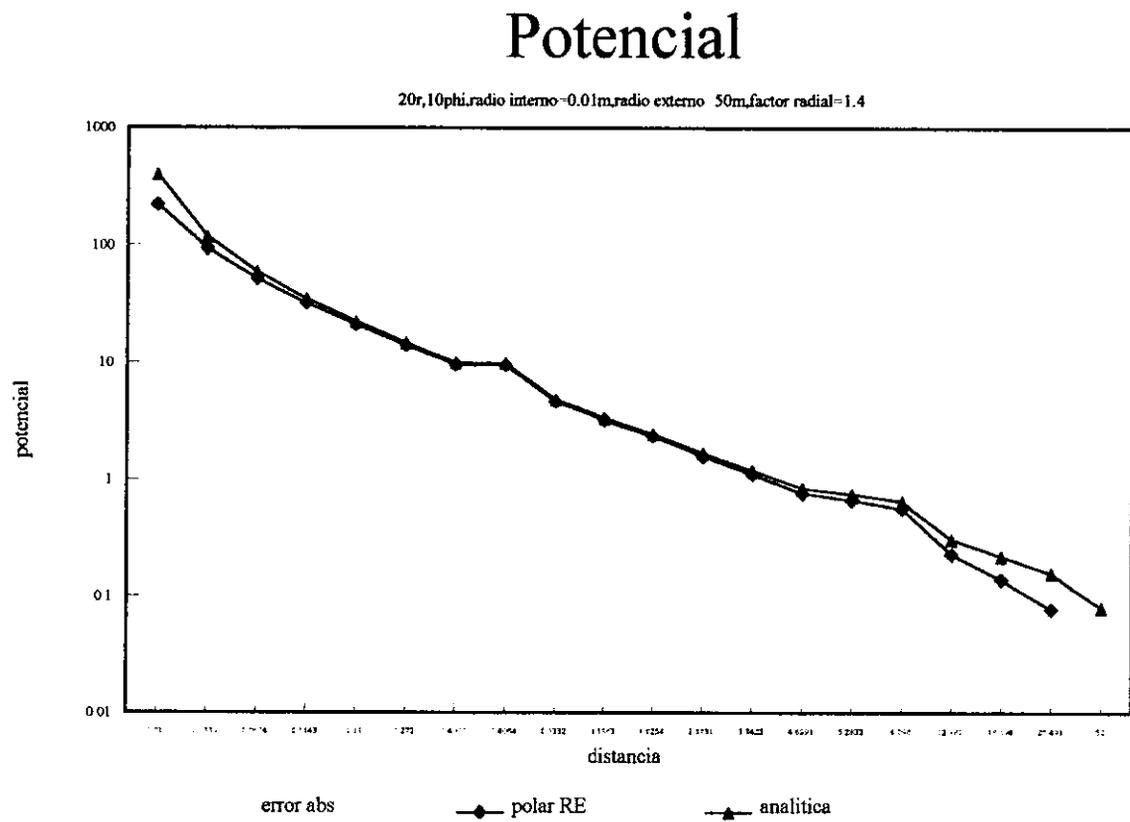


Figura 3.1: Malla esférica en medio homogéneo

Potencial

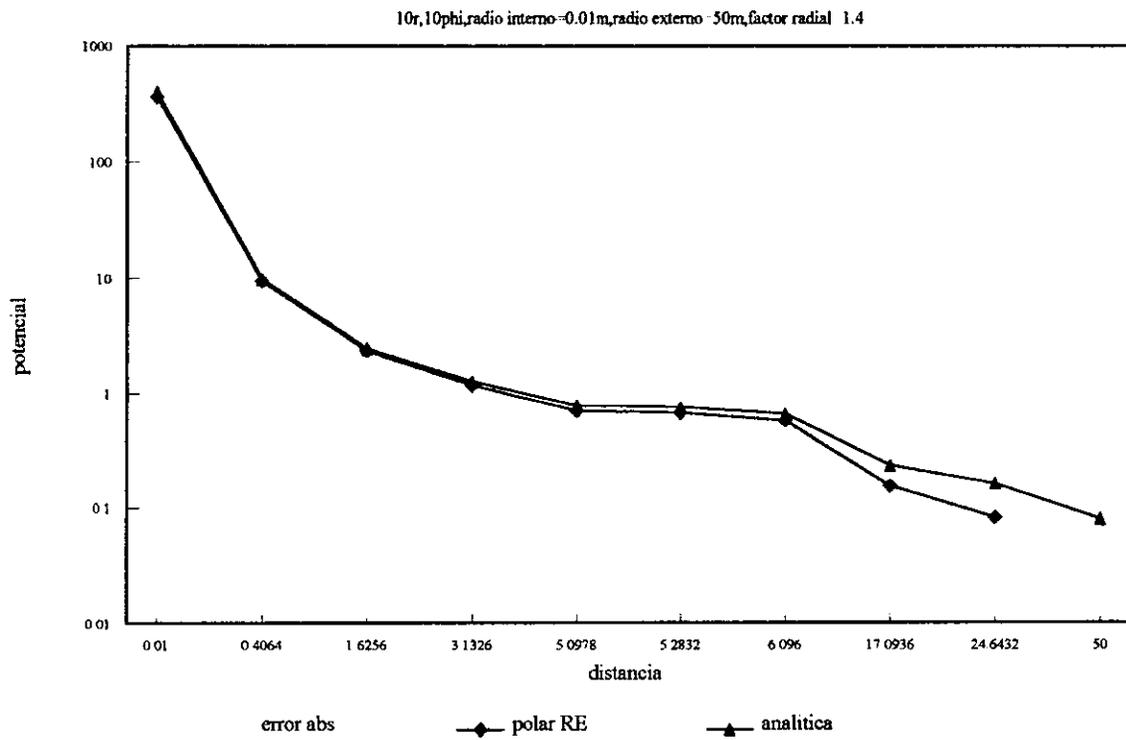


Figura 3.2: Disminución del factor de expansión radial a 1.4

Potencial

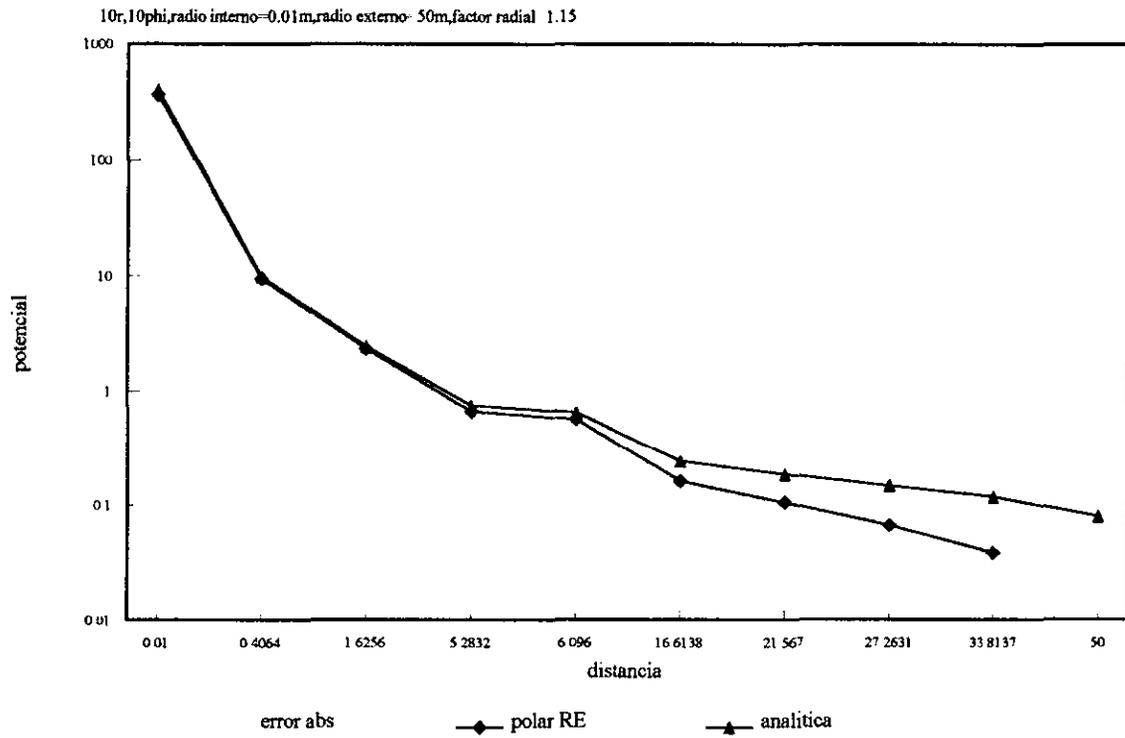


Figura 3.3: Disminución del factor de expansión radial a 1.15

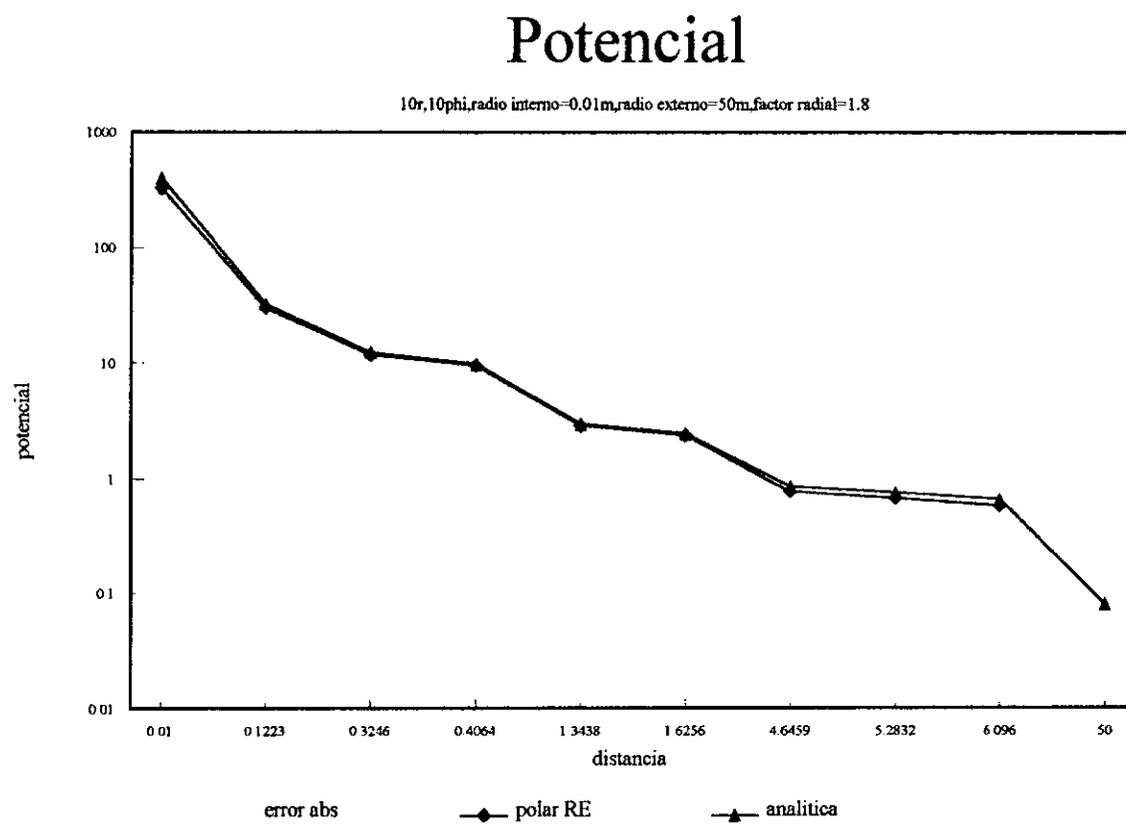


Figura 3.4: Aumento en el factor de expansión radial

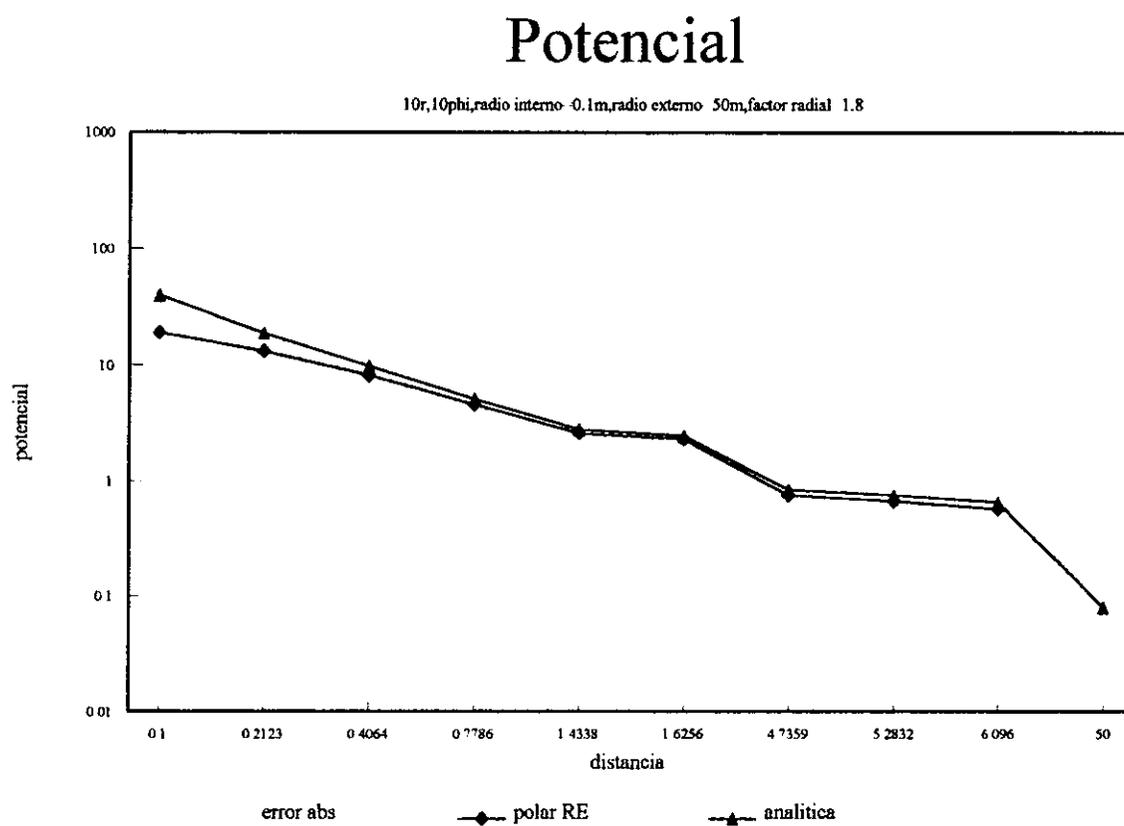


Figura 3.5: Radio interno de 0.1m

Potencial

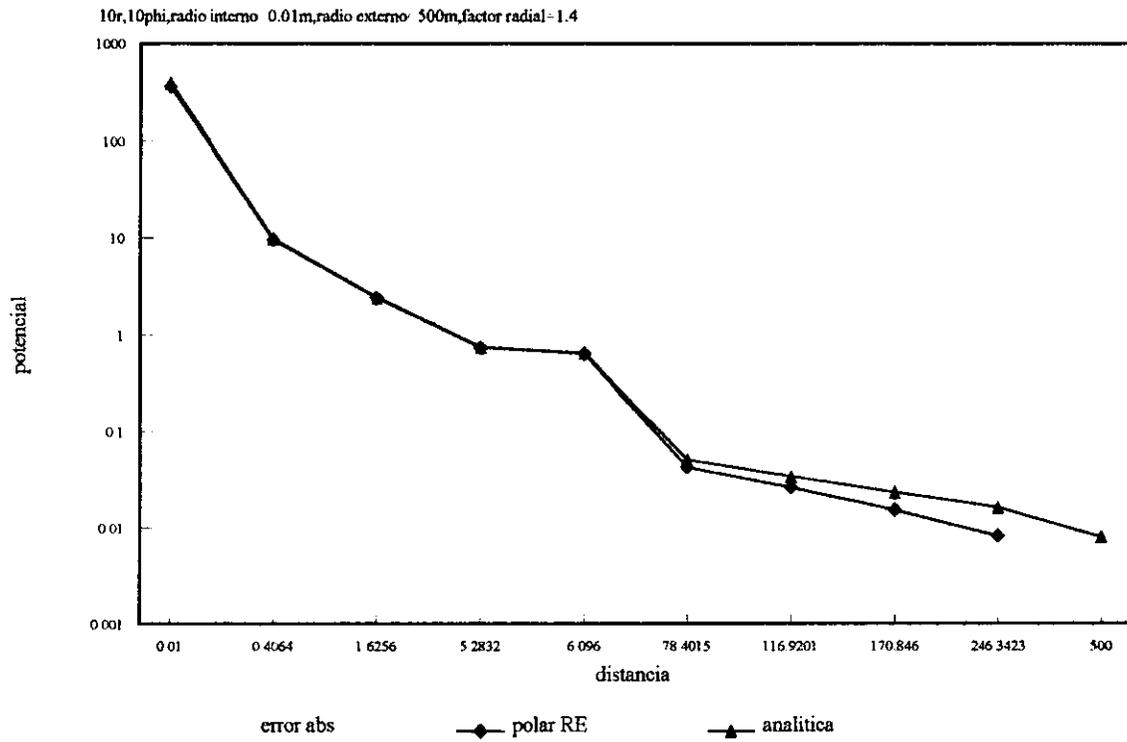


Figura 3.6: Incremento en el radio externo sin FRT

Potencial

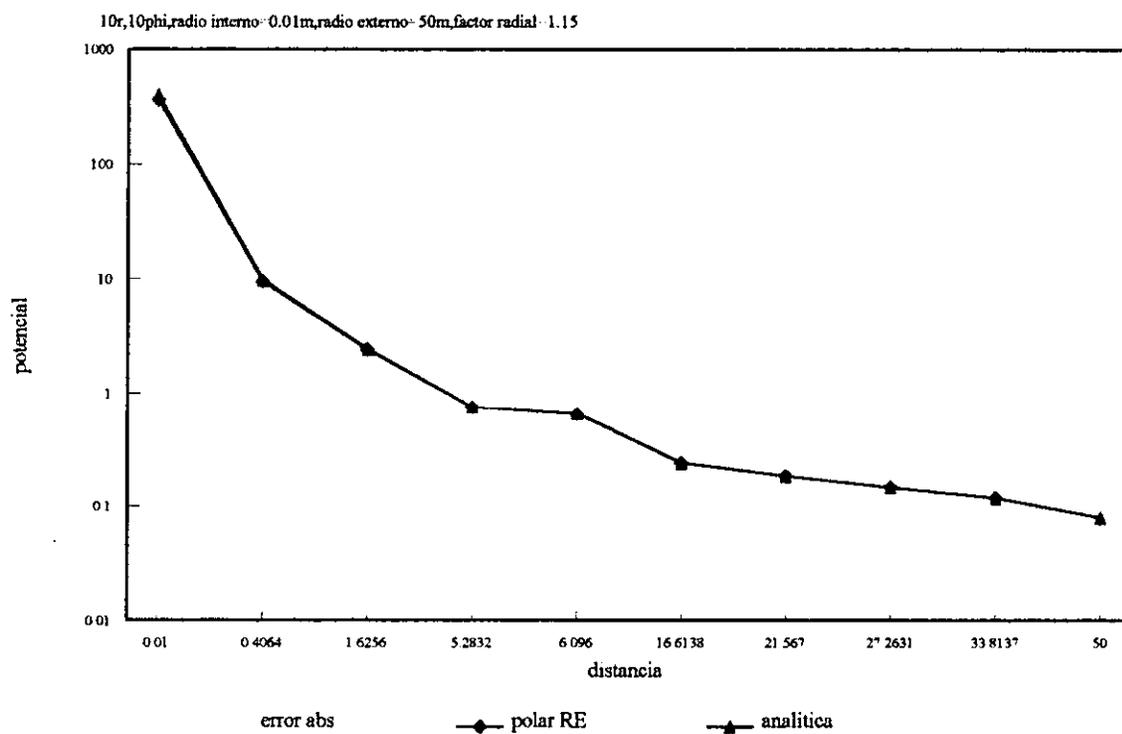


Figura 3.7: El factor de resistencia terminal

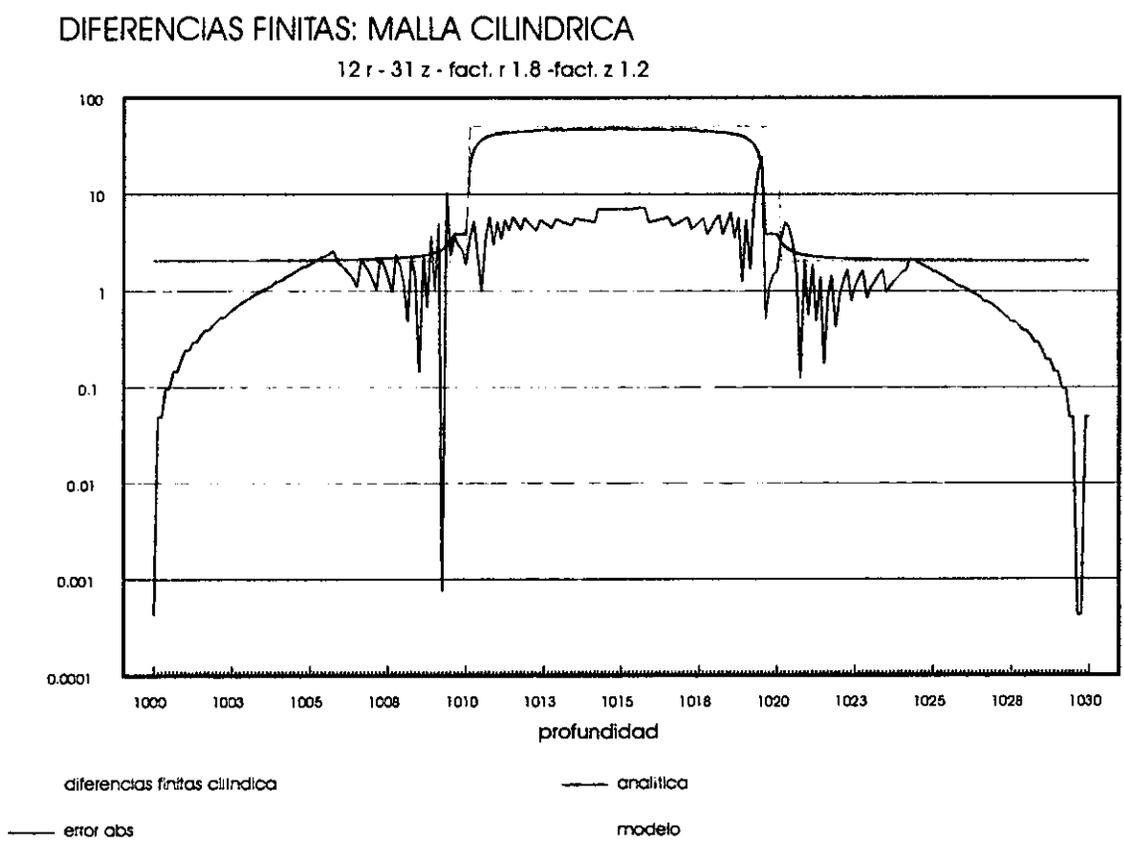
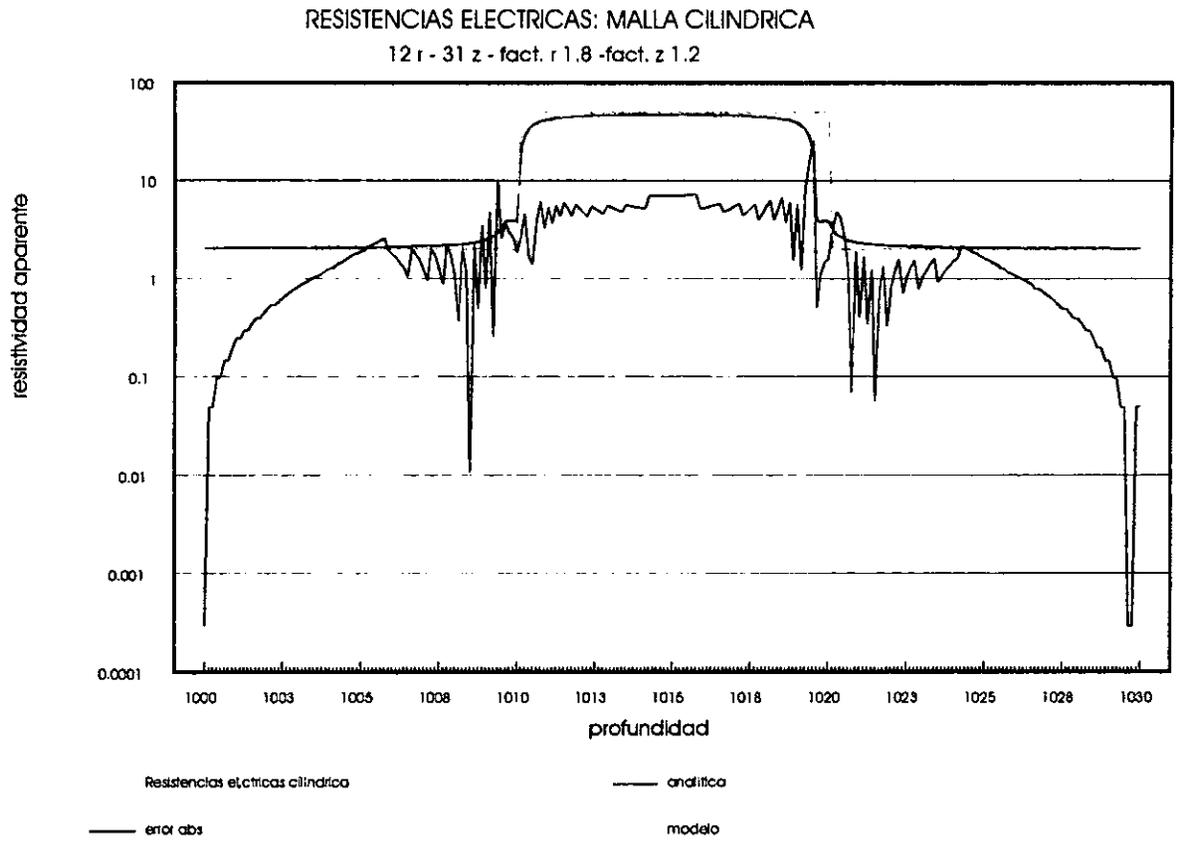


Figura 3.8: Malla cilíndrica

**error a la entrada
de una nueva capa
al introducir el nodo
de potencial a la malla**

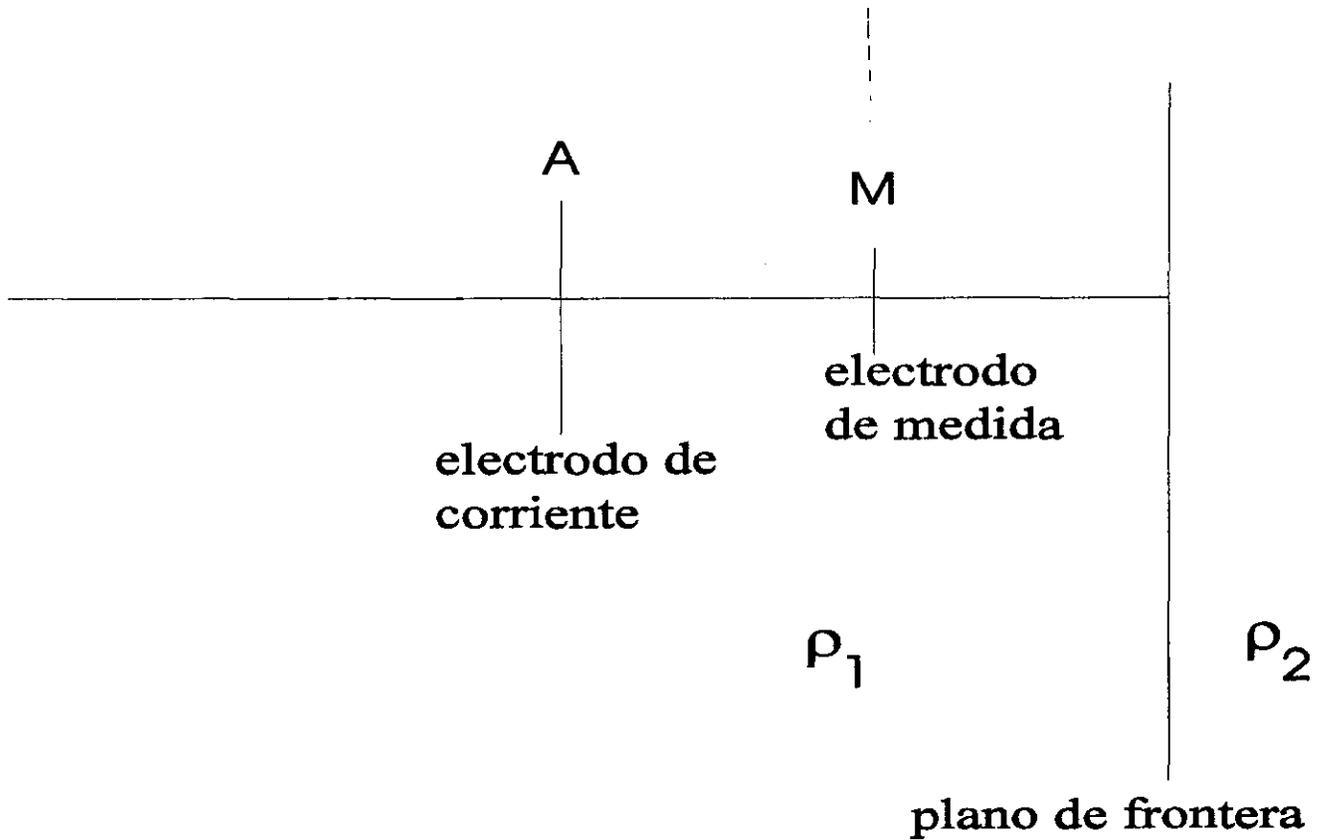
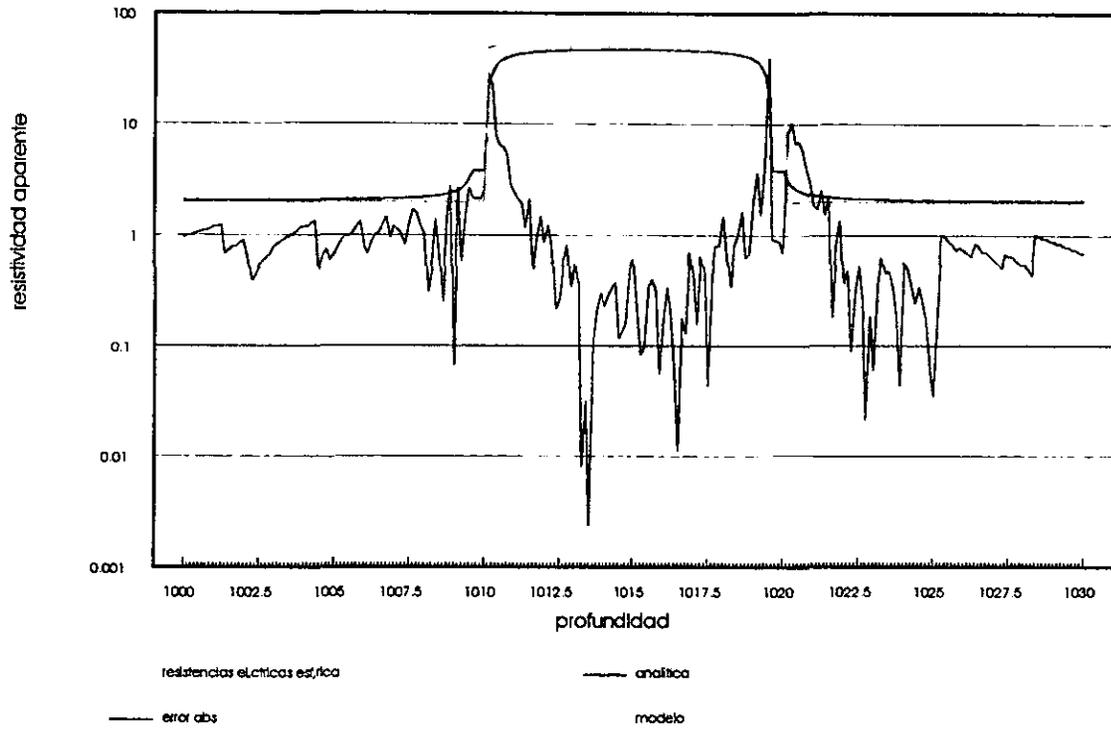


Figura 3.9: Error en la malla

RESISTENCIAS ELECTRICAS: MALLA ESFERICA

14 phi - 20 r -factor radial 1.26



87

DIFERENCIA FINITAS: MALLA ESFERICA

14 r - 20 phi -factor radial 1.24

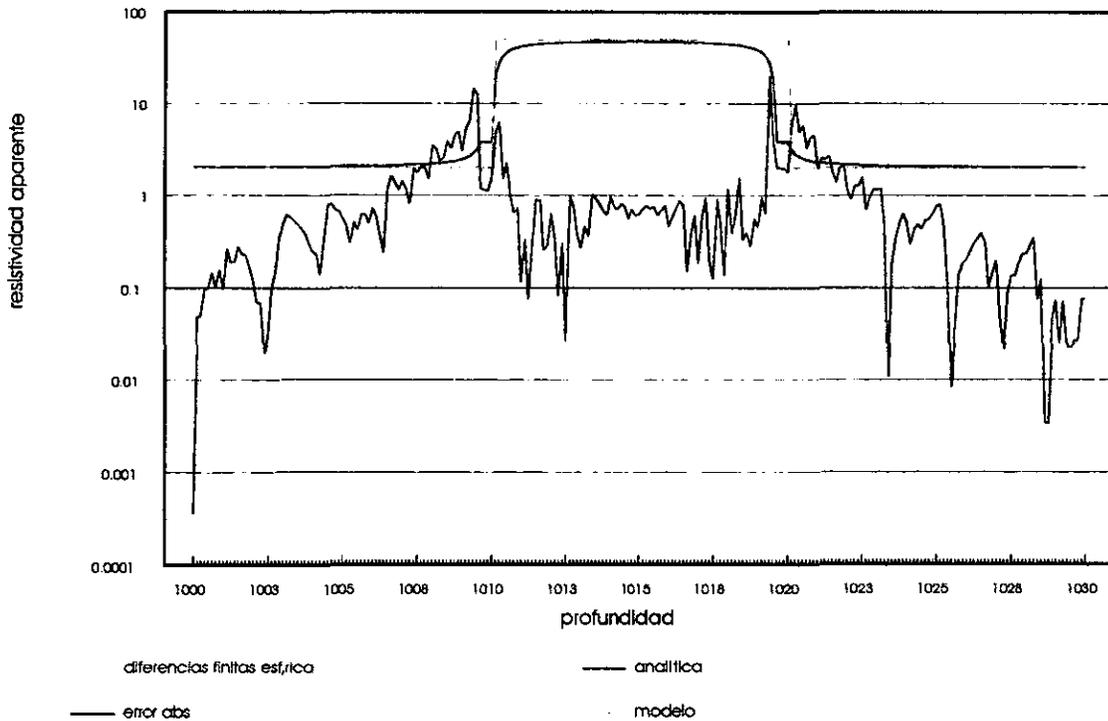
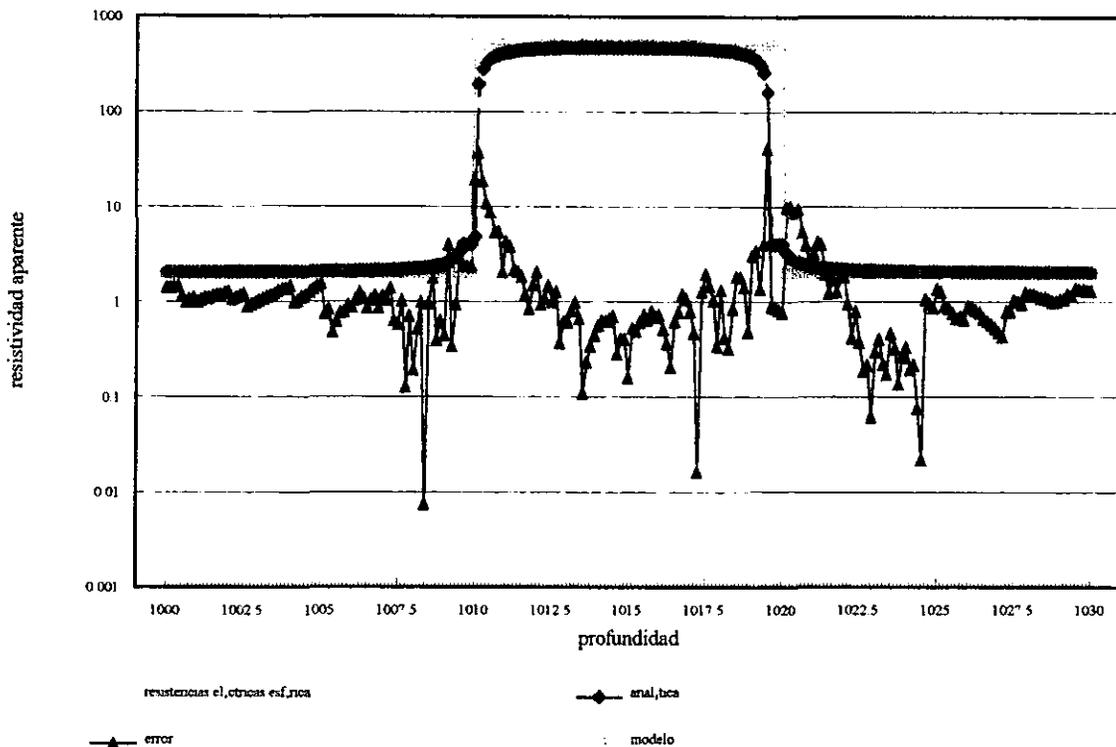


Figura 3.10: Malla Esférica

RESISTENCIAS ELECTRICAS: MALLA ESFERICA

12 phi - 20 r - fact. r 1.26



DIFERENCIAS FINITAS: MALLA CILINDRICA

12 r - 31 z - fact. r 1.8 - fact. z 1.2

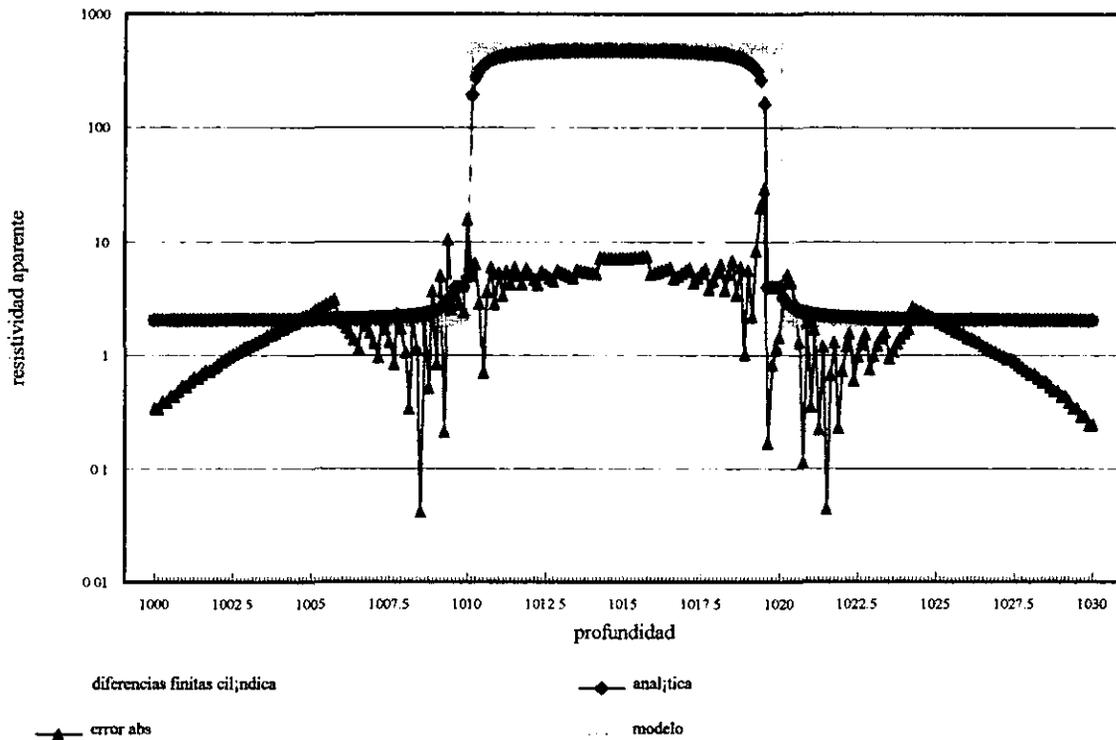
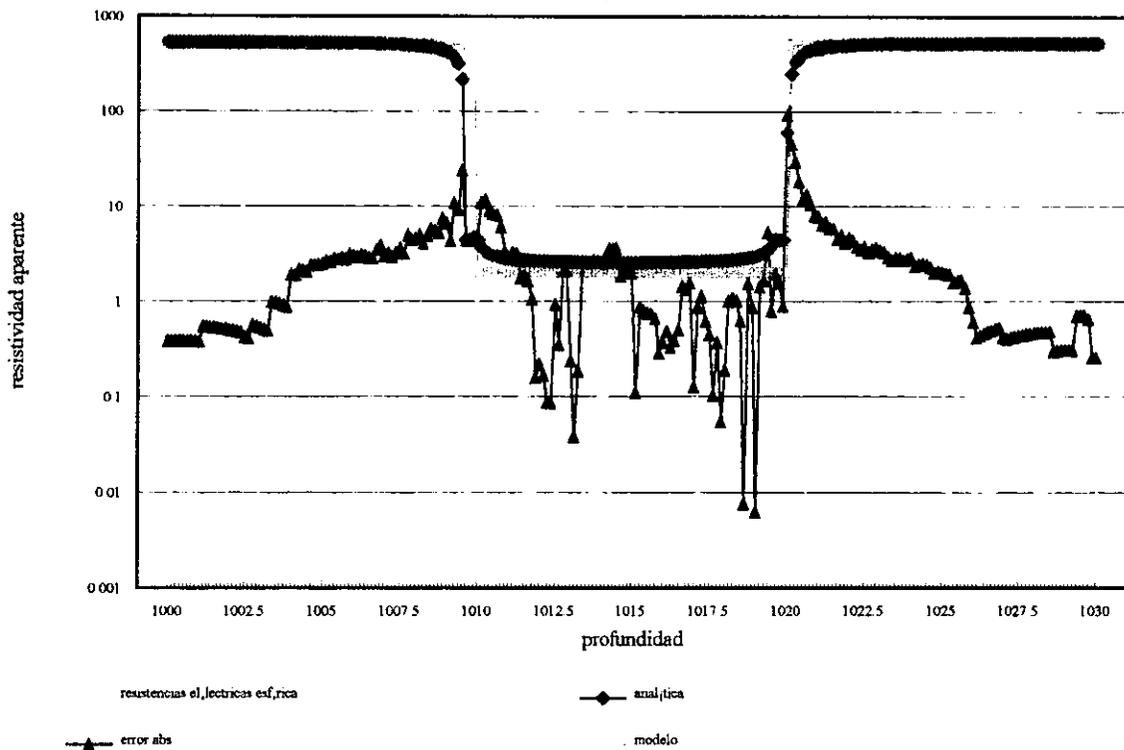


Figura 3.11: Contraste 2-500-2

RESISTENCIAS ELECTRICAS: MALLA ESFERICA

20 r -20 phi - fact. r 1.26



89

DIFERENCIAS FINITAS: MALLA CILINDRICA

12 r - 31 z - fact. r 1.8 - fact. z 1.2

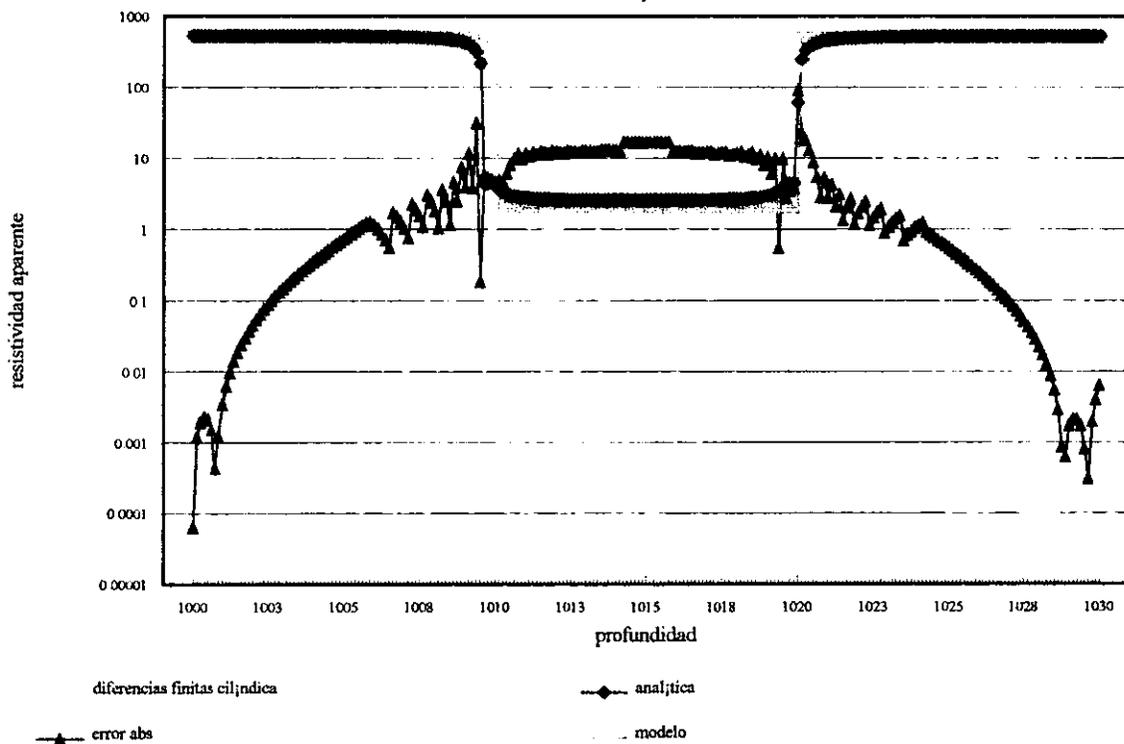
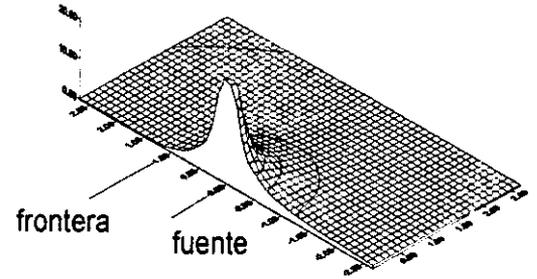
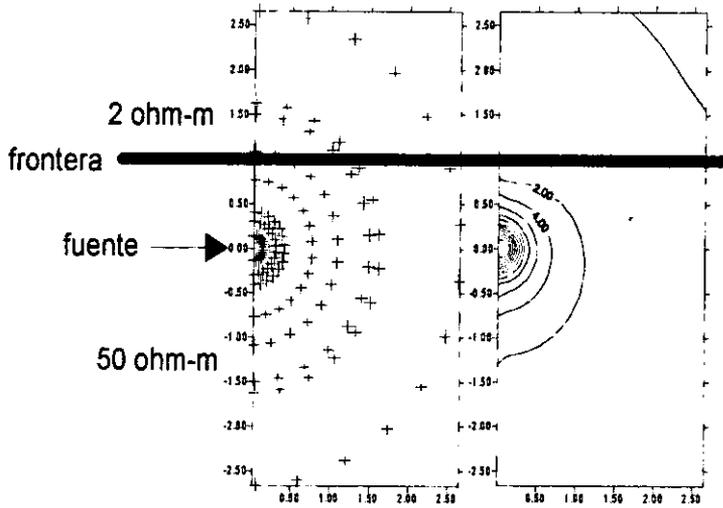


Figura 3.12: Contraste 500-2-500

potencial eléctrico: malla diferencias finitas esférica



potencial eléctrico: malla diferencias finitas cilíndrica

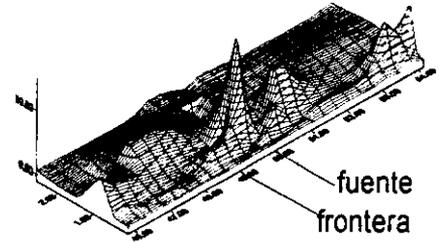
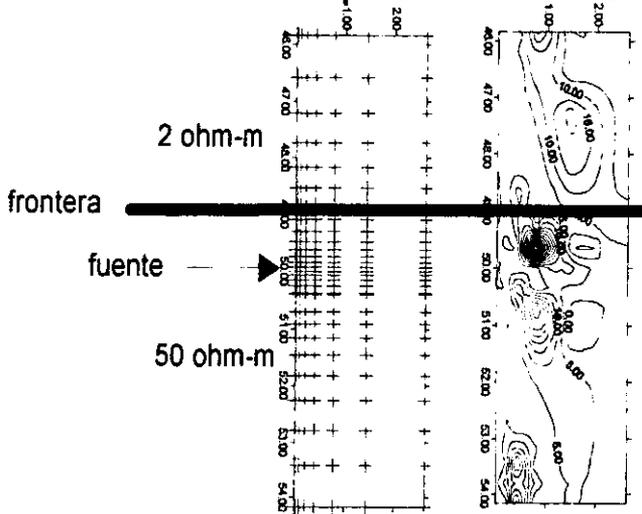


Figura 3.13: Contornos

Capítulo 4

Conclusiones

1.- Para iguales geometrías se tiene que la matriz de coeficientes en diferencias finitas es muy poco diferente a la matriz de coeficientes de resistencias eléctricas, debido por supuesto a que son diferentes aproximaciones para una solución única, aunque se pueda pensar que las resistencias eléctricas al ser obtenidas de los coeficientes en diferencias finitas debería de formar el mismo sistema de ecuaciones, no es así, y difieren en los parámetros para resolver la normal corta pero la variación casi es mínima, esto puede ser porque los elementos diferenciales de volumen utilizados en las resistencias eléctricas y en diferencias finitas no son exactamente los mismos.

2.- La geometría esférica, resultó como se esperaba, un método eficiente de determinar la curva de resistividad aparente de la normal corta con una disminución significativa en el número de nodos utilizados en comparación de la geometría cilíndrica, es la geometría y no el método numérico del que dependen los buenos resultados, esto claro para el valor MINIMO de nodos para capas horizontales paralelas.

3.- Es más fácil construir el programa cilíndrico porque las resistividades coinciden con la geometría de la malla, pero a cambio se requiere más tiempo de cómputo. Mientras que la esférica ofrece la posibilidad de reducir el número de nodos pero requiere más aplicación dar información a la malla esférica de los contrastes de resistividad de las capas horizontales

4- Los errores se presentan en las dos geometrías y aunque la esférica no es natural en medios horizontales en donde la cilíndrica ha existido siempre, las gráficas de

error muestran que puede ser utilizada como herramienta de modelado, se conoce la eficiencia de la cilíndrica en su medio, pero no se sabe como se comportará cuando se enfrente a medios no acorde con ella, por otro lado la esférica puede trabajar en medio no acorde con ella y podría ser posible su utilización para casos más complicados.

Capítulo 5

Apéndice

5.1 Medio con fronteras planas paralelas

Las ecuaciones para el potencial en un medio con planos paralelos puede ser obtenida por dos maneras: a) Por la integración directa de la ecuación de Laplace, o b) Por el método de las imágenes descrito por Thompson(1901).

La primera opción es más general, usándola, la función potencial puede ser encontrada para geometrías complejas, pero el método de las imágenes usado para casos simples de una y dos fronteras, muestra el significado físico de la función potencial mas claramente que la primera opción.

5.1.1 Un plano de frontera.

En el caso en donde un plano de frontera separa dos medios isotrópicos semi-infinitos de diferentes resistividades ρ_1 y ρ_2 , la función potencial puede ser obtenida por el método de las imágenes como sigue, figura 5.1:

1. En el medio 1, donde se coloca la fuente originalmente, el potencial es el mismo que si fuera un medio homogéneo isotrópico si se integra al electródo de corriente original, una segunda fuente de corriente ficticia colocada en el punto imagen de A a través del plano de separación P. La corriente para esta segunda fuente A', puede ser designada por I'.

2. En el medio 2, el potencial debe ser el mismo que si fuera un medio completo homogéneo isotrópico pero ajustando la corriente de la imagen A a I'' .

Si designamos las distancias de las fuentes de corriente A y A' a los puntos M_1 y M_2 donde el potencial se estudiará como R y R' respectivamente, se obtienen las expresiones:

Para el punto M_1 en el medio 1:

$$U_{M_1}^{(1)} = U_{M_1}^A + U_{M_1}^{A'} = \frac{\rho_1}{4\pi} \left(\frac{I}{R} + \frac{I'}{R'} \right)$$

que significa que el potencial en M_1 en el medio (1) es la suma de efectos del potencial en M_1 debido a las cargas A y A'.

mientras que para M_2

$$U_{M_2}^{(1)} = U_{M_2}^A = \frac{\rho_2 I''}{4\pi R}$$

En el contacto debe cumplirse:

$$U_{M_1}^{(1)} \Big|_{R=R'} = U_{M_2}^{(1)} \Big|_{R=R'}$$

quedando

$$\rho_1 (I + I') = \rho_2 I'' \quad (5.1)$$

y por otra oparte la continuidad de la densidad de corriente en el contacto

$$\frac{1}{\rho_1} \frac{\partial U_{M_1}^{(1)}}{\partial R} \Big|_{R=R'} = \frac{1}{\rho_2} \frac{\partial U_{M_2}^{(1)}}{\partial R} \Big|_{R=R'}$$

$$\frac{1}{4\pi} \left(-\frac{I}{R^2} \frac{dR}{dn} - \frac{I'}{R'^2} \frac{dR'}{dn} \right) = -\frac{I''}{4\pi R^2} \frac{dR}{dn}$$

pero

$$\frac{dR}{dn} = -\frac{dR'}{dn}$$

entonces

$$I - I' = I'' \quad (5.2)$$

resolviendo 5.1 y 5.2

$$I' = \left(\frac{\rho_2 - \rho_1}{\rho_2 + \rho_1} \right) I = k_{12} I$$

mientras que

$$I'' = \left(1 - \frac{\rho_2 - \rho_1}{\rho_2 + \rho_1} \right) I = \frac{2\rho_1}{\rho_2 + \rho_1} I$$

sustituyendo en los potenciales:

$$U_{M_1}^{(1)} = \frac{\rho_1 I}{4\pi} \left(\frac{1}{R} + \frac{k_{12}}{R'} \right)$$

y

$$U_{M_2}^{(1)} = \frac{\rho_2}{4\pi R} (1 - k_{12}) I = \frac{I}{4\pi R} \frac{2\rho_1 \rho_2}{\rho_2 + \rho_1}$$

pero al estar en coordenadas cartesianas con el origen del sistema en la fuente original A y siendo L la distancia entre el electrodo de corriente y de potencial asi como ϕ el ángulo entre el eje z y el vector de posición R, ademas siendo Z_1 , la distancia perpendicular de la fuente al plano y la imagen situada a $2Z_1$:

para la fuente en el medio 1

$$R = L$$

$$\mathbf{R}' = \mathbf{r} - \mathbf{r}' = -L \cos \phi \mathbf{k} + L \sin \phi \mathbf{j} - (-2Z_1 \mathbf{k})$$

$$R' = \sqrt{(-L \cos \phi + 2Z_1)^2 + (L \sin \phi)^2}$$

$$R' = \sqrt{L^2 \cos^2 \phi - 4Z_1 L \cos \phi + 4Z_1^2 + L^2 \sin^2 \phi}$$

$$R' = \sqrt{L^2 + 4Z_1^2 - 4Z_1 L \cos \phi}$$

y los potenciales

$$U_{M_1}^{(1)} = \frac{\rho_1 I}{4\pi} \left(\frac{1}{L} + \frac{k_{12}}{\sqrt{L^2 + 4Z_1^2 - 4Z_1 L \cos \phi}} \right)$$

$$U_{M_2}^{(1)} = \frac{I}{4\pi L} \frac{2\rho_1\rho_2}{\rho_2 + \rho_1};$$

Si la fuente de corriente esta en el medio 2

$$\mathbf{R}' = \mathbf{r} - \mathbf{r}' = -L \cos \phi \mathbf{k} + L \sin \phi \mathbf{j} - (2Z_1 \mathbf{k})$$

$$R' = \sqrt{L^2 + 4Z_1^2 + 4Z_1 L \cos \phi}$$

y siendo los potenciales

$$U_{M_1}^{(2)} = \frac{I}{4\pi L} \frac{2\rho_1\rho_2}{\rho_2 + \rho_1}$$

$$U_{M_2}^{(2)} = \frac{\rho_2 I}{4\pi} \left(\frac{1}{L} + \frac{k_{21}}{\sqrt{L^2 + 4Z_1^2 + 4Z_1 L \cos \phi}} \right);$$

$$K_{21} = \frac{\rho_1 - \rho_2}{\rho_1 + \rho_2} = -k_{12}$$

diferenciando las ecuaciones respecto a el espaciamento L se puede encontrar el campo eléctrico a lo largo del eje z

$$E_{M_1}^{(1)} = -\frac{\partial U_{M_1}^{(1)}}{\partial L} = \frac{\rho_1 I}{4\pi} \left(\frac{1}{L^2} + \frac{k_{12}(L - 2Z_1 \cos \phi)}{(L^2 + 4Z_1^2 - 4Z_1 L \cos \phi)^{3/2}} \right)$$

$$E_{M_2}^{(1)} = -\frac{\partial U_{M_2}^{(1)}}{\partial L} = \frac{I}{4\pi L^2} \frac{2\rho_1\rho_2}{\rho_2 + \rho_1};$$

$$E_{M_2}^{(2)} = -\frac{\partial U_{M_2}^{(2)}}{\partial L} = \frac{\rho_2 I}{4\pi} \left(\frac{1}{L^2} + \frac{k_{21}(L + 2Z_1 \cos \phi)}{(L^2 + 4Z_1^2 + 4Z_1 L \cos \phi)^{3/2}} \right)$$

$$E_{M_1}^{(2)} = -\frac{\partial U_{M_1}^{(2)}}{\partial L} = \frac{I}{4\pi L^2} \frac{2\rho_1\rho_2}{\rho_2 + \rho_1};$$

Sonda potencial

a. para los electrodos A y M en el medio 1

$$\rho_{appa,M_1}^{(1)} = 4\pi L \frac{U_{M_1}^{(1)}}{I} = \rho_1 \left(1 + \frac{k_{12}}{\sqrt{1 + 4(Z_1/L)((Z_1/L) - \cos \phi)}} \right)$$

b. Para el electrodo A en el medio 1 y el electrodo M en el medio 2

$$\rho_{appa,M_2}^{(1)} = 4\pi L \frac{U_{M_2}^{(1)}}{I} = \frac{2\rho_1\rho_2}{\rho_1 + \rho_2}$$

c. Para el electrodo A en el medio 2 y el electrodo M en 1

$$\rho_{appa,M_1}^{(2)} = 4\pi L \frac{U_{M_1}^{(2)}}{I} = \frac{2\rho_1\rho_2}{\rho_1 + \rho_2}$$

d. Para los electrodos A y M en el medio 2

$$\rho_{appa,M_2}^{(2)} = 4\pi L \frac{U_{M_2}^{(2)}}{I} = \rho_2 \left(1 + \frac{k_{21}}{\sqrt{1 + 4(Z_1/L)((Z_1/L) + \cos \phi)}} \right)$$

Sonda gradiente

a. Con los electrodos A, M y N situados en el medio 1

$$\rho_{appa,M_1}^{(1)} = 4\pi L^2 \frac{E_{M_1}^{(1)}}{I} = \rho_1 \left(1 + \frac{K_{12}(1 - 2(Z_1/L) \cos \phi)}{(1 + 4(Z_1/L)((Z_1/L) - \cos \phi))^{3/2}} \right)$$

b. con el electrodo A en el medio 1 y los electrodos M y N en el medio 2

$$\rho_{appa,M_2}^{(1)} = 4\pi L^2 \frac{E_{M_2}^{(1)}}{I} = \frac{2\rho_1\rho_2}{\rho_1 + \rho_2}$$

c. Con el electrodo A en el medio 2 y electrodos M y N en el medio 1

$$\rho_{appa,M_1}^{(2)} = 4\pi L^2 \frac{E_{M_1}^{(2)}}{I} = \frac{2\rho_1\rho_2}{\rho_1 + \rho_2}$$

d. Con todos los electrodos en el medio 2

$$\rho_{appa,M_2}^{(2)} = 4\pi L^2 \frac{E_{M_2}^{(2)}}{I} = \rho_2 \left(1 + \frac{K_{21}(1 + 2(Z_1/L) \cos \phi)}{(1 + 4(Z_1/L)((Z_1/L) + \cos \phi))^{3/2}} \right)$$

En el caso de $\phi = 0^\circ$ o sea el eje del pozo penetrando verticalmente a la

capa:

Los dos electrodos en el medio 1

Para la sonda potencial

$$\rho_{appa,M_1}^{(1)} = \rho_1 \left(1 + \frac{k_{12}}{2(Z_1/L) - 1} \right)$$

Para la sonda gradiente

$$\rho_{appa,M_1}^{(1)} = \rho_1 \left(1 - \frac{k_{12}}{(2(Z_1/L) - 1)^2} \right)$$

Debe conservarse el mismo sistema de referencia para obtener resultados coherentes.

Si solo el electrodo A esta en un medio y los demás en el otro se tiene la misma ecuación para ambas sondas

$$\rho_{appa,M_2}^{(1)} = \frac{2\rho_1\rho_2}{\rho_1 + \rho_2} = \rho_{appa,M_1}^{(2)}$$

Con todos los electrodos en el segundo medio

sonda potencial

$$\rho_{appa,M_2}^{(2)} = \rho_2 \left(1 + \frac{k_{21}}{2(Z_1/L) + 1} \right)$$

sonda gradiente

$$\rho_{appa,M_2}^{(2)} = \rho_2 \left(1 + \frac{k_{21}}{(2(Z_1/L) + 1)^2} \right)$$

5.1.2 Dos fronteras

Utilizando el método de las imagenes se puede obtener la solución a la ecuación de Laplace. El problema es encontrar la función potencial para tres regiones designadas como 1,2 y 3 con resistividades ρ_1, ρ_2 y ρ_3 , las regiones estan separadas por dos planos paralelos, P y Q. El sistema de referencia esta centrado en el electrodo de corriente A, las capas estan en la parte negativa del sistema y el eje Z intersecta las capas horizontales perpendicularmente, el electrodo de medición forma un angulo ϕ con el eje vertical, las distancia entre los planos es h, figura 5.2 .

Si tenemos una corriente I en el electrodo A del medio 1 a una distancia Z_1 del plano P, entonces se tendrá su imagen en el medio 2 a una distancia $2Z_1$, como sabemos del análisis de una frontera esta imagen tiene la magnitud $I' = k_{12}I$. Con esto se satisface la condición de frontera P, asumiendo también que la función potencial en el medio 2 se encuentra si se ajusta la intensidad de corriente de A a $(1 - k_{12})I$, esto también del análisis de una sola frontera, ahora se debe satisfacer la frontera Q.

Entonces el potencial en el medio 1 debido a la corriente I_3 de la imagen en 3 y la última corriente encontrada $I_1 = (1 - k_{12})I$ de A será:

$$U_1^{(1)} = \frac{\rho_1}{4\pi} \left(\frac{I_1}{R} + \frac{I_3}{R_3} \right)$$

mientras que para un punto de medición $A_1^{(3)}$ imagen en 3

$$U_2^{(1)} = \frac{\rho_3 I_4}{4\pi R}$$

donde I_4 se refiere a la más actual corriente ajustada de A, aplicando las condiciones de frontera a potenciales y derivadas:

$$\rho_1(I_1 + I_3) = \rho_3 I_4$$

$$I_1 - I_3 = I_4$$

quedando

$$I_3 = k_{23} I_1 = I(1 - k_{12})k_{23}$$

$$k_{23} = \frac{\rho_3 - \rho_2}{\rho_3 + \rho_2}$$

$$I_4 = I_1 - I_3 = (1 - k_{23})I_1 = I(1 - k_{12})(1 - k_{23})$$

siendo $R_3 = \sqrt{L^2 + 4(Z_1 + h)^2 - 4(Z_1 + h) \cos \phi}$

pero es necesario equilibrar a la frontera P, así que se debe encontrar la magnitud de la imagen de $I(1 - k_{12})k_{23}$, en el medio 1, es como si solo existieran 1 y

2 pero la intensidad de corriente es la de 3. Si $I_5 = I(1 - k_{12})k_{23}$, y I_6 es la intensidad de corriente a determinar en 1 a una distancia $z=2h+z$

$$\rho_2(I_5 + I_6) = \rho_1 I_4$$

$$I_5 - I_6 = I_4$$

$$I_6 = k_{21} I_5 = I(1 - k_{12})k_{23}k_{21}$$

pero esta nueva corriente no satisface a la frontera Q, así que hay que ajustar en el medio 3 la corriente, o sea de 1 a 3 con ρ_2 en la fuente de 1

$I_7 = I(1 - k_{12})k_{23}k_{21}$ en medio 1, I_8 por determinar en el medio 3

$$\rho_2(I_7 + I_8) = \rho_3 I_4$$

$$I_7 - I_8 = I_4$$

$$I_8 = k_{23} I_7 = I(1 - k_{12})k_{23}k_{21}k_{23}$$

localizada en el punto del medio 3

$$z = 2(2h + Z)$$

Pero como es lógico, es necesario encontrar el reflejo de esta en el medio 1 respecto del plano P

a una distancia $z=-4h$

repetimos el mismo proceso de encontrar la imagen en 1 de 3 como si estuviese el medio 2 completo, sin 3

$I_9 = I(1 - k_{12})k_{23}k_{21}k_{23}$, en 3 con ρ_2 y I_{10} por determinar en 1 con ρ_1

$$I_{10} = k_{21} I_9 = I(1 - k_{12})k_{23}^2 k_{21}^2$$

que a su vez producirá su imagen respecto del plano Q a una distancia $z=3(2h+Z)$

$$I_{11} = k_{23}I(1 - k_{12})k_{23}^2k_{21}^2k_{23}$$

que continuando con el proceso, se puede ver que el potencial en el medio 1 debido a una fuente de corriente en el mismo 1, depende de una sola imagen en el medio 2 más la suma infinita de imagenes en el medio 3

$$U_1^{(1)} = \frac{\rho_1 I}{4\pi} \left(\frac{1}{L} + \frac{k_{12}}{\sqrt{L^2 + 4Z^2 - 4ZL \cos \phi}} + \right. \\ \left. + (1 - k_{12})k_{23} \sum_{n=1}^{\infty} \frac{(k_{21}k_{23})^{n-1}}{\sqrt{((n2h) + 2Z)^2 + L^2 - 2((n2h) + 2Z)L \cos \phi}} \right)$$

El potencial en el medio 2 debido a la fuente de corriente en 1, será la sumatoria de las corrientes en el medio 1 mas la sumatoria de las corrientes imagen en 3

$$U_2^{(1)} = \frac{\rho_2 I}{4\pi} \left((1 - k_{12}) \sum_{n=0}^{\infty} \frac{(k_{21}k_{23})^n}{\sqrt{(2nh)^2 + L^2 + 2(2nh)L \cos \phi}} + \right. \\ \left. + (1 - k_{12})k_{23} \sum_{n=1}^{\infty} \frac{(k_{21}k_{23})^{n-1}}{\sqrt{((n2h) + 2Z)^2 + L^2 - 2((n2h) + 2Z)L \cos \phi}} \right)$$

Para la fuente de corriente en el medio 1 y el potencial en 3, se toman las sumatorias de las imagenes solo en el medio 1, debido a la corriente exitadora inicial de $I(1 - k_{12})(1 - k_{23})$

$$U_3^{(1)} = \frac{\rho_3 I}{4\pi} (1 - k_{12})(1 - k_{23}) \sum_{n=0}^{\infty} \frac{(k_{12}k_{23})^n}{\sqrt{(2nh)^2 + L^2 + 2(2nh)L \cos \phi}}$$

Si la fuente de corriente esta en el medio 2 a una distancia Z del plano P, necesitamos una imagen en $z=-2Z$ en el medio 1, de intensidad Ik_{21} . La misma carga original es reflejada por Q a una distancia $z=2(h-z)$, de intensidad Ik_{23} .

Se inicia el proceso de encontrar el reflejo de estas dos intensidades de corriente respecto a los planos, los reflejos de $I_{k_{23}}$ son:

en 1

$$\sum_{n=1}^{\infty} \frac{(k_{21}k_{23})^n}{\sqrt{(2nh)^2 + L^2 + 2(2nh)L \cos \phi}}$$

en 3

$$k_{23} \sum_{n=1}^{\infty} \frac{(k_{21}k_{23})^{n-1}}{\sqrt{(2nh - 2Z)^2 + L^2 - 2(2nh - 2Z)L \cos \phi}}$$

Los reflejos de $I_{k_{21}}$

en 1

$$k_{21} \sum_{n=0}^{\infty} \frac{(k_{21}k_{23})^n}{\sqrt{(2nh + 2Z)^2 + L^2 + 2(2nh + 2Z)L \cos \phi}}$$

en 3

$$\sum_{n=1}^{\infty} \frac{(k_{21}k_{23})^n}{\sqrt{(2nh)^2 + L^2 - 2(2nh)L \cos \phi}}$$

quedando

$$U_1^{(2)} = \frac{\rho_2 I}{4\pi} \left(\frac{1}{L} + \sum_{n=1}^{\infty} \frac{(k_{21}k_{23})^n}{\sqrt{(2nh)^2 + L^2 + 2(2nh)L \cos \phi}} + \right. \\ \left. + k_{23} \sum_{n=1}^{\infty} \frac{(k_{21}k_{23})^{n-1}}{\sqrt{(2nh - 2Z)^2 + L^2 - 2(2nh - 2Z)L \cos \phi}} + \sum_{n=1}^{\infty} \frac{(k_{21}k_{23})^n}{\sqrt{(2nh)^2 + L^2 - 2(2nh)L \cos \phi}} + \right. \\ \left. + k_{21} \sum_{n=0}^{\infty} \frac{(k_{21}k_{23})^n}{\sqrt{(2nh + 2Z)^2 + L^2 + 2(2nh + 2Z)L \cos \phi}} \right)$$

que para la normal corta queda:

Para la fuente de corriente y electrodo de potencial en el medio 1, la primer frontera esta en $Z=0$

$$\rho_{apa}^{(1)}(Z, L, h, k_{12}, k_{23}, \rho_1) = \rho_1 \left(1 + \frac{Lk_{12}}{\sqrt{L^2 + 4Z^2 + 4ZL \cos \phi}} + \right.$$

$$+(1 - k_{12})k_{23}L \sum_{n=1}^{\infty} \frac{(k_{12}k_{23})^{n-1}}{\sqrt{(2nh + 2Z)^2 + L^2 - 2(2nh + 2Z)L \cos \phi}} \quad Z \leq L$$

Para la fuente y electrodo en el medio 2

$$\begin{aligned} \rho_{apa}^{(2)}(Z, L, h, k_{21}, k_{23}, \rho_2) = & \rho_2 \left(1 + L \sum_{n=1}^{\infty} \frac{(k_{21}k_{23})^n}{\sqrt{(2nh)^2 + L^2 + 2(2nh)L \cos \phi}} + \right. \\ & + Lk_{23} \sum_{n=1}^{\infty} \frac{(k_{21}k_{23})^{n-1}}{\sqrt{(2nh - 2Z)^2 + L^2 - 2(2nh - 2Z)L \cos \phi}} + L \sum_{n=1}^{\infty} \frac{(k_{21}k_{23})^n}{\sqrt{(2nh)^2 + L^2 - 2(2nh)L \cos \phi}} + \\ & \left. + Lk_{21} \sum_{n=0}^{\infty} \frac{(k_{21}k_{23})^n}{\sqrt{(2nh + 2Z)^2 + L^2 + 2(2nh + 2Z)L \cos \phi}} \right) \quad 0 \leq Z \leq h - Z \text{ y } L \leq h \end{aligned}$$

Para el electrodo de corriente en 1 y de potencial en 2

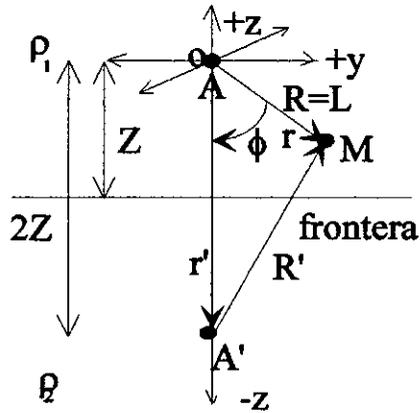
$$\begin{aligned} \rho_{apa}^{(21)}(Z, h, L, k_{21}, k_{23}, \rho_2) = & \rho_2 L (1 - k_{12}) \left(\sum_{n=0}^{\infty} \frac{(k_{21}k_{23})^n}{\sqrt{(2nh)^2 + L^2 + 2(2nh)L \cos \phi}} + \right. \\ & \left. + k_{23} \sum_{n=1}^{\infty} \frac{(k_{21}k_{23})^{n-1}}{\sqrt{(2nh + 2Z)^2 + L^2 - 2(2nh + 2Z)L \cos \phi}} \right) \quad \text{para } -L \leq Z \leq 0 \end{aligned}$$

no es necesario otras ecuaciones pues por simetría se obtiene: electrodo de corriente en 2 y electrodo de potencial en 3

$$\rho_{apa}^{(32)}(Z, L, h, k_{12}, k_{23}, \rho_2) = \rho_{apa}^{(21)}(Z - h + L, h, -k_{23}, -k_{12}) \quad \text{para } Z \leq h - L \text{ y } Z < h$$

Para los dos electrodos en 3

$$\rho_{apa}^{(3)}(Z, L, h, k_{12}, k_{23}) = \frac{\rho_3}{\rho_1} \rho_{apa}^{(1)}(Z - h + L, h, -k_{23}, -k_{12}) \quad \text{para } Z \geq h$$



sistema de referencia

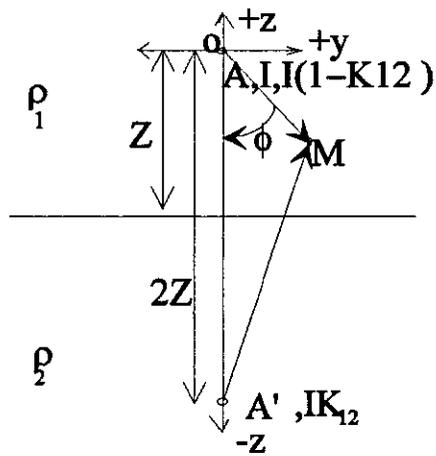
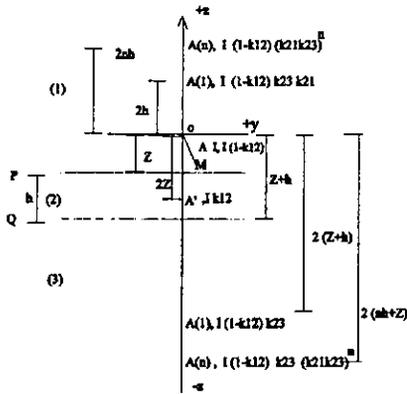
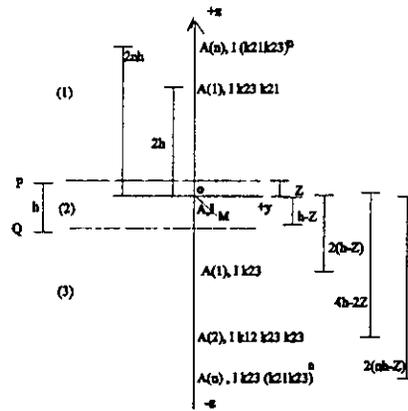


imagen para una frontera

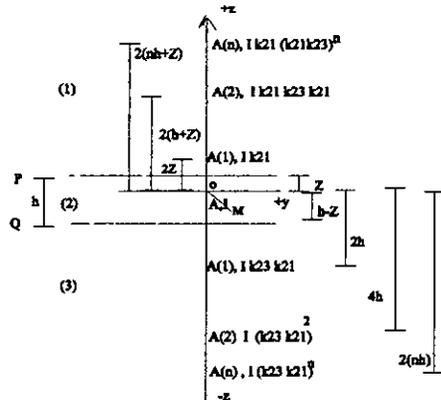
Figura 5.1: Analítica frontera simple



fuente en el medio 1



fuente en el medio 2
reflejos de Ik23



fuente en el medio 2
reflejos de Ik21

Figura 5.2: Analítica dos fronteras

Bibliografía

- [1] H.Spurlin y Anguiano Rojas 1997 Complexities of the Lateral Response and Techniques That Facilitate ES-Log Interpretation.SPWLA 38th Annual Logging Symposium. June 15-18
- [2] Dakhnov,V.N.,1962,Geophysical Well Logging.Quarterly of the Colorado School of Mines. Vol 57 ,No 2
- [3] Kim, J.,H.1986. Forward Modeling and Inversion of Responses for Borehole Normal and Lateral Electrode Arrangements. Ph.D. diss. Colorado School of Mines.187 pp.
- [4] Anguiano Rojas, 1991, Inversion of Conventional Oil Well Petrophysical Logs. Master of Science Thesis, Standford University.
- [5] Mufti, I.R., 1980 , Finite-Difference Evaluation of Apparent Resistivity Curves. Geophysical Prospecting, Vol. 28, pp 146-166.
- [6] Orellana E.,1982 , Prospección Geoeléctrica en Corriente Continua. 2a ed. Ed. Paraninfo, madrid 1982
- [7] Shoichiro, N. 1992, Métodos Numéricos Aplicados con Software. 1a. ed, Ed. Prentice-Hall Hispanoamericana, Cap 11, p. 405-450.
- [8] Curtis, F. G. 1991, Análisis Numérico. 2a. ed, Ed. Alfaomega, cap. 7, p. 388-429,
- [9] Taylor, R.J., 1982, An Introduction to Error Analysis.1a ed, Oxford University Press.

- [10] Director, W. 1975, Circuit Theory, A computational Approach , John Wiley, BoulderColorado
- [11] Corbeiller, P.Le.,1950, Matrix Analysis of Electric Networks, John wiley and Sons. Inc.
- [12] Kron Gabriel, 1938, Application of Tensors to The analysis of Rotating Electrical Machinery. General Electric Review, Shenectady, Part XVI.
- [13] Kaufman A.A. 1992, Geophysical Field Theory and Method, Part A. Academic Press inc.
- [14] Wangsness R.,K., 1993, Campos Electromagnéticos, Ed. Limusa, México.
- [15] McQuistan B.R, 1985, Campos Escalares y Vectoriales, Ed Limusa, México.
- [16] Gardiol Fred, 1987, Curso Intermedio de Electromagnetismo, Ed Limusa, México.
- [17] Reitz R,J 1969, Fundamentos de la Teoría Electromagnética, Ed.UTEHA, México.
- [18] Hayt H.W.,1991, Teoría Electromagnética. 2a. ed, Ed. McGraw-Hill,México.

Anexo

Programas

Programa de diferencias finitas cilíndrico

```

/*Programa de diferencias finitas cilindrica
 * Ricardo Zavala Torres
 */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#include "nrutil.h"
#define NR_END 0
#define FREE_ARG char*
#define P3 atan(1)*4

double *tool;
double *R2;
double *PHI;
double **OHR;
double **W;
double **S;
double yy,zz;
double *abd;
double *d1C;
double *dbC;
double *dcC;
double *b,*x;
double *lay;
double *Z1;
double *profabs;
double *OHR1;
int *OHR2;
int *tooli;

void nrerror(char error_text[])
/* Numerical Recipes standard error handler */
{
    fprintf(stderr,"Numerical Recipes run-time error...\n");
    fprintf(stderr,"%s\n",error_text);
    fprintf(stderr,"...now exiting to system...\n");
    exit(1);
}

int *ivector(long nl, long nh)
/* allocate an int vector with subscript range v[nl..nh] */
{
    int *v;

    v=(int *)malloc((size_t) ((nh-nl+1+NR_END)*sizeof(int)));
    if (!v) nrerror("allocation failure in ivector()");
    return v-nl+NR_END;
}

double *dvector(long nl, long nh)
/* allocate a double vector with subscript range v[nl..nh] */
{
    double *v;

    v=(double *)malloc((size_t) ((nh-nl+1+NR_END)*sizeof(double)));
    if (!v) nrerror("allocation failure in dvector()");
    return v-nl+NR_END;
}

double **dmatrix(long nrl, long nrh, long ncl, long nch)
/* allocate a double matrix with subscript range m[nrl..nrh][ncl..nch] */
{
    long i, nrow=nrh-nrl+1, ncol=nch-ncl+1;
    double **m;

    /* allocate pointers to rows */
    m=(double **) malloc((size_t)((nrow+NR_END)*sizeof(double*)));
    if (!m) nrerror("allocation failure 1 in matrix()");
    m += NR_END;
    m -= nrl;
}

```

```

/* allocate rows and set pointers to them */
m[nrl]=(double *) malloc((size_t)((nrow*ncol+NR_END)*sizeof(double)));
if (!m[nrl]) perror("allocation failure 2 in matrix()");
m[nrl] += NR_END;
m[nrl] -= ncl;

for(i=nrl+1;i<=nrh;i++) m[i]=m[i-1]+ncol;

/* return pointer to array of pointers to rows */
return m;
}

void free_ivector(int *v, long nl)
/* free an int vector allocated with ivector() */
{
    free((FREE_ARG) (v+nl-NR_END));
}

void free_dvector(double *v, long nl)
/* free a double vector allocated with dvector() */
{
    free((FREE_ARG) (v+nl-NR_END));
}

void free_dmatrix(double **m, long nrl, long ncl)
/* free a double matrix allocated by dmatrix() */
{
    free((FREE_ARG) (m[nrl]+ncl-NR_END));
    free((FREE_ARG) (m+nrl-NR_END));
}

int max0C(int num1, int num2)
{
    int regreso;
    if(num1 >= num2)
        regreso=num1;
    else
        regreso=num2;

    return(regreso);
}

int min0C(int num1, int num2)
{
    int regreso;
    if(num1 <= num2)
        regreso=num1;
    else
        regreso=num2;

    return(regreso);
}

double ddotC(int n, int la, int k, int N, int mu, int j)
{
    int i;
    double regreso;

    regreso=0;
    for(i=0; i<n; i++)
        regreso+=abd[(la+i)*N+k]*abd[(mu+i)*N+j];
    return(regreso);
}

double ddot2C(int n, int la, int k, int N, double *dy)
{
    int i;
    double regreso;

```

```

regreso=0;
for(i=0;i<n;i++)
{
    regreso+=abd[(la+i)*N+k]*(dy);
    dy++;
}
return(regreso);
}

void daxpyC(int n,double da,int la,int k,int N, double *dy)
{
    int m,i,mp1;
    double *dyanterior;

    dyanterior=dy;

    m=n%4;
    for(i=1;i<=m;i++)
    {
        *dy=*dy+da*abd[(la+i-1)*N+k];
        dy++;
    }

    dy=dyanterior;
    if(n>=4)
    {
        mp1=m+1;
        dy+=m;
        for(i=mp1;i<=n;i++)
        {
            *dy=*dy+da*abd[(la+i-1)*N+k];
            dy++;
        }
    }
}

void adaptC(int jmax,int kmax)
{
    int i,k0,j,k,ik,jk,mu;
    int la,lm,lb,kb;
    int m,n,jmax2;
    double s,t;
    FILE *arch;

    /*write the b vector to be solved*/
    /* b=(double *)propio_malloc((kmax+1)*sizeof(double));

    for(i=1;i<=kmax;i++)
        b[i]=0;
    k0=(jmax+1)*(i0-1)+1;
    b[k0]=current;*/

    /*Band Storage*/
    m=jmax+1;
    jmax2=jmax+2;
    n=kmax;

    abd=dvector(0,(jmax2+1)*(n+1));

    /*zero the matrix*/
    for(j=1;j<=n;j++)
    {
        for(i=1;i<=jmax2;i++)
            abd[i*n+j]=0;
    }
}

```

```

/*write in the non zero diagonals in band format*/
for(i=1;i<=n;i++)
abd[jmax2*n+i]=d1C[i];
for(i=2;i<=n;i++)
abd[m*n+i]=dbC[i-1];
for(i=jmax2;i<=n;i++)
abd[l*n+i]=dcC[i-m];

/*do the L D L factoring*/

for(j=1;j<=n;j++)
{
s=0;
ik=m+1;
jk=max0C(j-m,1);
mu=max0C(m+2-j,1);
if(m>=mu)
{
for(k=mu;k<=m;k++)
{
t=0;
t=abd[k*n+j]-ddotC(k-mu,ik,jk,n,mu,j);
t=t/abd[(m+1)*n+jk];
abd[k*n+j]=t;
s=s+(t*t);
ik--;
jk++;
}
}
/*aqui significa que m fue menor a mu */
s=abd[(m+1)*n+j]-s;
abd[(m+1)*n+j]=sqrt(s);
}/*j*/

/*solve trans(R)*y=b*/
for(k=1;k<=n;k++)
{
lm=min0C(k-1,m);
la=m+1-lm;
lb=k-lm;
t=ddot2C(lm,la,k,n,&b[lb]);
b[k]=(b[k]-t)/abd[(m+1)*n+k];
}

/*solve r*x=y*/

for(kb=1;kb<=n;kb++)
{
k=n+1-kb;
lm=min0C(k-1,m);
la=m+1-lm;
lb=k-lm;
b[k]=b[k]/abd[(m+1)*n+k];
t=b[k]*-1;
daxpyC(lm,t,la,k,n,&b[lb]);
}
free_dvector(abd,0);
}
void malla(int I1,int J1,double F1,double F2,double PHI1,double RH1,double R1)
{
int i,j,k;
double D1,D11,D2,zz;

R2=dvector(0,J1);
PHI=dvector(0,I1);
tooli=ivector(0,5);
Z1=dvector(0,I1);

```

```

if (F1 == 1.0)

        D1=PHI1/(I1-1);
        else
        {
                D1=(PHI1)*(1-F1)/(1-pow(F1,I1));
                D11=(PHI1)*(1-F1)/(1-pow(F1,I1));
        }

if (F2 == 1.0)

        D2=RH1/(J1-1);
        else
        D2=RH1*(1-F2)/(1-pow(F2,J1));

R2[0]=R1;
/*printf("%d,%6.2fn",0,R2[0]);*/
for(j=1;j<J1;j++)
{
        R2[j]=R2[j-1]+D2;
        D2=F2*D2;

        /* printf("%d,%6.2fn" j,R2[j]); */
}

/*for(j=0;j<J1;j++)
{
        if (fabs(R2[j]- tool)<0.07 ) R2[j]=tool;
        printf("%d,%6.4fn" j,R2[j]);
}
*/

/* for(j=0;j<J1;j++)
{
        if (R2[j]<tool) R2[j+1]=tool;
        R2[J1-1]=R2H1;
        printf("%d,%6.4fn" j,R2[j]);
        fprintf(fp,"%d,%6.4fn" j,R2[j]);
}*/

for (k=0;k<1;k++)
for(j=0;j<J1;j++)
{
        if (R2[j]<tool[k] && R2[j+1]>tool[k] ) R2[j+1]=tool[k];

        /* printf("%d,%6.4fn" j,R2[j]);
        fprintf(fp,"%d,%6.4fn" j,R2[j]);*/
}
R2[J1-1]=RH1;
/*PHI[0]=0;
printf("%d,%6.2fn",0,PHI[0]);
for(i=1;i<I1;i++)
{
        PHI[i]=PHI[i-1]+D1;
        D1=F1*D1;
        printf("%d,%6.2fn",i,PHI[i]);
} */

PHI[0]=0.0;
PHI[(I1-1)/2]=PHI1/2;
for(i=(I1-1)/2-1;i>0;i--)
{
        PHI[i]=PHI[i+1]-D1;
        D1=F1*D1;
        /*printf("%d,%6.2f,%6.2fn",i,PHI[i],D1);
        fprintf(fp,"%d,%6.2f,%6.2fn",i,PHI[i],D1);*/
}

```

```

for(i=(I1-1)/2+1;i<I1;i++)
    {
        PHI[i]=PHI[i-1]+D11;
        D11=F1*D11;
    }
PHI[I1-1]=PHI1;
for (k=0;k<1;k++)
for (i=(I1-1)/2+1;i<I1;i++)
    {
        zz=(PHI1/2+tool[k]);
        if (PHI[i]<zz && PHI[i+1]>zz)
            { PHI[i+1]=(PHI1/2)+tool[k];
              tooli[k]=i+1;
            }
    }

for(i=0;i<I1;i++)
    {
        Z1[i]=PHI1/2-PHI[i];
    }
}
void xyrho1(int I1,int J1,int nlay,float sourceprof)
{ int i,j,k;

  OHR=dmatrix(0,I1+1,0,J1+1);
  profabs=dvector(0,nlay+1);

for (i=0;i<nlay-1;i++)
    {
        profabs[i]=(sourceprof-lay[i]);
    }

for (k=0;k<nlay-1;k++)
    {
for(j=0;j<=J1;j++)
for(i=0;i<=I1;i++)
    {
        if (profabs[k]==Z1[i])
            OHR[i][j]=OHR1[k];
        if (profabs[k]>Z1[i] && profabs[k+1]<Z1[i])
            OHR[i][j]=OHR1[k+1];

        if (profabs[0]<Z1[i])
            OHR[i][j]=OHR1[0];
        if (profabs[nlay-2]>Z1[i])
            OHR[i][j]=OHR1[nlay-1];
    }
    }
}
}

```

```

void matriz(int I1,int J1,int ni)
{ int i,j,k;
float o,p,q,y,yy,yyy,z,zz,V;

b=dvector(0,ni+1);
d1C=dvector(0,ni+1);
dcC=dvector(0,ni+1);
dbC=dvector(0,ni+1);
x=dvector(0,ni+1);

for (i=1;i<I1-1;i++)
{
for(j=0;j<J1-1;j++)

{

d1C[(i+j)+(J1-2)*(i-1)]=0.0;
dcC[(i+j)+(J1-2)*(i-1)]=0.0;
dbC[(i+j)+(J1-2)*(i-1)]=0.0;

}
}

for (i=1;i<I1-1;i++)
{
j=0;
o=(PHI[i]);
p=(PHI[i+1]);
q=(PHI[i-1]);
y=R2[0]*((R2[1]-R2[0]));
z=(p-o)*(p-q);
zz=(o-q)*(p-q);

V=2*P3*R2[0]*((R2[1]-R2[0])/2)*((p-q)/2);

d1C[(i+j)+(J1-2)*(i-1)]=V*W[i][0]*((2/pow(R2[1]-R2[0],2))+(1/y)))
+(V*S[i][j]*2/z)
+(V*S[i-1][j]*2/zz);
dbC[(i+j)+(J1-2)*(i-1)]=-V*W[i][0]*((2/pow(R2[1]-R2[0],2))+(1/y)));

dcC[(i+j)+(J1-2)*(i-1)]=-V*S[i][j]*(2/z);
if(i==I1-2)
dcC[(i+j)+(I1-3)*j]=0.0;
}

for (i=1;i<I1-1;i++)
{
for(j=1;j<J1-1;j++)

{

o=(PHI[i]);
p=(PHI[i+1]);
q=(PHI[i-1]);
y=R2[j]*(R2[j+1]-R2[j-1]);
yy=(R2[j+1]-R2[j-1])*(R2[j+1]-R2[j]);
yyy=(R2[j+1]-R2[j-1])*(R2[j]-R2[j-1]);

z=(p-o)*(p-q);
zz=(o-q)*(p-q);
}
}
}

```

```

V=2*P3*R2[j]*((R2[j+1]-R2[j-1])/2)*((p-q)/2);

d1C[(i+j)+((J1-2)*(i-1))]=(V*W[i][j]*((2/yy)+(1/y))
+(V*W[i][j-1]*((2/yyy)-(1/y)))
+(V*S[i][j]*2/z)
+(V*S[i-1][j]*2/zz);

dbC[(i+j)+((J1-2)*(i-1))]=-V*W[i][j]*((2/yy)+(1/y));

dcC[(i+j)+((J1-2)*(i-1))]=-V*S[i][j]*(2/z);
if(j==J1-2)
  dbC[(i+j)+((J1-2)*(i-1))]=0.0;
if(i==I1-2)
  dcC[(i+j)+((J1-2)*(i-1))]=0.0;

}
}

/*for(i=0;i<ni;i++)
{
  for(j=0;j<ni;j++)
  {
a[i*ni+j]=0;
if(i==j) a[i*ni+j]=-A[i]*V[i];
if((i+j)==(I1-2)+(i*2))a[i*ni+j]=-B[i]*V[i];
if((i+j)==1+(i*2))a[i*ni+j]=-C[i]*V[i];
if((i+j)==(I1-2)+(j*2))a[i*ni+j]=-B[j]*V[j];
if((i+j)==1+(j*2))a[i*ni+j]=-C[j]*V[j];
printf(fp,"%10.5f",a[i*ni+j]);
}
printf(fp,"\n");
}*/

for (i=1;i<=ni;i++)
{

  b[i]=0.0;
  x[i-1]=0;
  if(i==((I1-1)/2+1+(J1-2)*((I1-1)/2-1)) {b[i]=1;
  x[i-1]=1;}

/*printf("%3.2fn",b[i]);
fprintf(fp,"%3.2fn",b[i]);*/
/* fprintf(fp,"%d,%10.5f,%10.5f,%10.5f,%10.5fn",i,b[i],d1C[i],dbC[i],dcC[i]);*/
/*printf("%d,%10.5f,%10.5f,%10.5f,%10.5fn",i,b[i],d1C[i],dbC[i],dcC[i]);*/
}

}

void xyrho(int I1,int J1,int nlay,float PHI1,float sourceprof)

{ int i,j,k;
OHR=dmatrix(0,I1+1,0,J1+1);
Z1=dvector(0,nlay+1);
profabs=dvector(0,nlay+1);
OHR2=ivector(0,nlay+1);

```

```

for (i=0;i<nlay-1;i++)
{
    profabs[i]=(sourceprof-lay[i]);

    if (profabs[i]>0)
        Z1[i]=PHI1/2-fabs(profabs[i]);
    else
        Z1[i]=PHI1/2+fabs(profabs[i]);
}

for (k=0;k<nlay-1;k++)
{
for(i=0;i<=I1;i++)
{
    if (PHI[i]<Z1[k] && PHI[i+1]>Z1[k])
    {
        PHI[i]=Z1[k];
        OHR2[k]=i;
    }
    else
        if (PHI[i]==Z1[k])
        {
            /*PHI[i]=Z1[k];*/
            OHR2[k]=i; }
}
}

for (i=0;i<=I1;i++)
{
    for (j=0;j<=J1;j++)
    {
        OHR[i][j]=OHR1[nlay-1];
    }
}

for(k=0;k<nlay-1;k++)
{
    for (i=0;i<=OHR2[1-k];i++)
    {
        for (j=0;j<=J1;j++)
            OHR[i][j]=OHR1[1-k];
    }
}

}
main()
{

int I1,J1,i,j,n,zj,ni,1,k,m,t,ip,pi,N,aa,bb,nlay;
float RH1,R1,PHI1,PHI2,PHI3,F1,F2,D1,D11,D2,A1,A2,A3,A4,OHR0,R6,R3,TANPHLM1,C3,D3;
float z,w,v,y,sumaa,suma,sum,smax,r1,c,sumaaa,dphi,dr;
float o,p,q,r,analytical_error,xx,yyy,start,sourceprof,kk,end,samp,res,fro;
/* start to set geometry and grid */

char name[14];
FILE *fp;
printf("Archivo a salvar:");
gets(name);
if (name[0]!='\0'){
fprintf(stderr,"no filename typed\n");
exit(EXIT_SUCCESS);
}
}

```

```

fp=fopen(name,"w+");
if (fp==NULL){
fprintf(stderr,"cannot open %s\n",name);
exit(EXIT_FAILURE);
}

tool=dvector(0,5);

printf("number of nodes in r direction:\n");
scanf("%d",&J1);
printf("number of nodes in phi direction:\n");
scanf("%d",&I1);
fprintf(fp,"number of nodes in r direction: %d\n",J1);
fprintf(fp,"number of nodes in phi direction: %d\n",I1);

/* RH1 is the radius of dona and PHI1 is the angle of calculation zone */

/* internal radius of dona */
R1=0;

printf("internal radius of cylinder:\n");
scanf("%f",&R1);
printf("radial thickness of cylinder:\n");
scanf("%f",&RH1);
printf("length of cylinder:\n");
scanf("%f",&PHI2);
fprintf(fp,"internal radius of cylinder: %f\n",R1);
fprintf(fp,"radial thickness of cylinder: %f\n",RH1);
fprintf(fp,"length of cylinder: %f\n",PHI2);

/*PHI1=PHI2;*/
/* *P3/180, */

/* F1 and F2 expansion factors in r and phi direction */

printf("Enter Grid-expansion factor in radial:\n");
scanf("%f",&F2);
printf("Enter Grid-expansion factor in z:\n");
scanf("%f",&F1);
fprintf(fp,"Enter Grid-expansion factor in radial: %f\n",F2);
fprintf(fp,"Enter Grid-expansion factor in z: %f\n",F1);

/*J1=12;
I1=31;
R1=0.01;
RH1=50;
PHI2=100;
F1=1.2;
F2=1.8;*/
PHI1=PHI2;
ni=(J1-1)*(I1-2);
tool[0]=0.4064;
tool[1]=1.6256;
tool[2]=5.2832;
tool[3]=6.0960;

printf("Enter number of layers:\n");
scanf("%d",&nlay);
fprintf(fp,"Enter number of layers: %d\n",nlay);

```

```

/*nlay=3;*/
lay=dvector(0,nlay+1);
OHR1=dvector(0,nlay+1);
/*start=1000;*/
printf("start:\n");
scanf("%f",&start);
fprintf(fp,"start: %f\n",start);
printf("end:\n");
scanf("%f",&end);
fprintf(fp,"end: %f\n",end);
printf("samp rate:\n");
scanf("%f",&samp);
fprintf(fp,"samp rate: %f\n",samp);

/*
lay[0]=1010;
lay[1]=1020;
lay[2]=10060;
OHR1[0]=0.002;
OHR1[1]=0.5;
OHR1[2]=0.002;
OHR1[3]=0.05;
*/
for(i=0;i<nlay;i++)
{
    printf("Enter resistivity of layer %d:\n",i+1);
    scanf("%f",&res);
    fprintf(fp,"resistivity of layer %d: %f\n",i+1,res);
    OHR1[i]=1/res;
}
for(k=0;k<nlay-1;k++)
{
    printf("Enter depth of boundary %d:\n",k+1);
    scanf("%f",&fro);
    fprintf(fp,"depth of boundary %d: %f\n",k+1,fro);
    lay[k]=fro;
}

malla(I1,J1,F1,F2,PHI1,RH1,R1);

/* sourceprof=1009.250; */
/*printf("prof:");
scanf("%f",&sourceprof);*/

for(kk=0;kk<=240;kk++)
{
    sourceprof=start+(kk*0.125);

xyrho(I1,J1,nlay,sourceprof);

/*xyrho(I1,J1,nlay,PHI1,sourceprof);*/

W=dmatrix(0,I1,0,J1);
S=dmatrix(0,I1,0,J1);

for (i=0;i<I1;i++)
{
    for (j=0;j<J1;j++)
    {
        W[i][j]=(2*(OHR[i][j]*OHR[i][j+1]))/(OHR[i][j]+OHR[i][j+1]);
        S[i][j]=(2*(OHR[i][j]*OHR[i+1][j]))/(OHR[i][j]+OHR[i+1][j]);
    }
}
}

```



```
i=tooli[0]-1;
j=0;
z=.4064*4*P3*b[i+j+(J1-2)*i]*1.15498;
    printf("%6.3f %12.6fn",sourceprof,z);
fprintf(fp,"%6.3f %12.6fn",sourceprof,z);
```

```
free_dmatrix(W,0,0);
free_dvector(b,0);
```

```
free_dmatrix(S,0,0);
free_dvector(profab,0);
free_dmatrix(OHR,0,0);
free_dvector(dbC,0);
free_dvector(d1C,0);
free_dvector(dcC,0);
free_dvector(x,0);
```

```
    }
free_dvector(OHR1,0);
free_dvector(R2,0);
free_dvector(PHI,0);
free_dvector(lay,0);
free_dvector(tool,0);
free_ivector(tooli,0);
free_dvector(Z1,0);
```

```
fclose(fp);
/* return EXIT_SUCCESS;*/
```

```
    }
```

Programa de diferencias finitas esférico

```

/*Programa de Diferencias finitas esférica
 * Ricardo Zavala Torres
 */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <alloc.h>
#include "mutil.h"
#define NR_END 0
#define FREE_ARG char*
#define P3 atan(1)*4

double *y;
double *Z;
double *R;
double *PHIT1;
double *PHIT2;
double *OHR;
double *OHR5;
double *W;
double *S;
double yy,zz;

double *abd;
double *d1C;
double *dbC;
double *dcC;
double *b;
double *tool;
int *jpot;
double *lay;

int max0C(int num1,int num2)
{
int regreso;
if(num1>=num2)
regreso=num1;
else
regreso=num2;

return(regreso);
}

int min0C(int num1,int num2)
{
int regreso;
if(num1<=num2)
regreso=num1;
else
regreso=num2;

return(regreso);
}

double ddotC(int n, int la,int k,int N,int mu,int j)
{
int i;
double regreso;

regreso=0;
for(i=0;i<n;i++)
regreso+=abd[(la+i)*N+k]*abd[(mu+i)*N+j];
return(regreso);
}

```

```

double ddot2C(int n,int la, int k, int N,double *dy)
{
  int i;
  double regreso;

  regreso=0;
  for(i=0;i<n;i++)
  {
    regreso+=abd[(la+i)*N+k]*(*dy);
    dy++;
  }
  return(regreso);
}

void daxpyC(int n,double da,int la,int k,int N, double *dy)
{
  int m,i,mp1;
  double *dyanterior,

  dyanterior=dy,

  m=n%4;
  for(i=1;i<=m;i++)
  {
    *dy=*dy+da*abd[(la+i-1)*N+k];
    dy++;
  }

  dy=dyanterior;
  if(n>=4)
  {
    mp1=m+1;
    dy+=m;
    for(i=mp1;i<=n;i++)
    {
      *dy=*dy+da*abd[(la+i-1)*N+k];
      dy++;
    }
  }
}

void adaptC(int jmax,int kmax)
{
  int i,k0,j,k,ik,jk,mu;
  int la,lm,lb,kb;
  int m,n,jmax2;
  double s,t;
  FILE *arch;

  /*write the b dvector to be solved*/
  /* b=(double *) malloc((kmax+1)*sizeof(double));

  for(i=1;i<=kmax;i++)
  b[i]=0;

  k0=(jmax+1)*(i0-1)+1;
  b[k0]=1; */

  /*Band Storage*/
  m=jmax+1;
  jmax2=jmax+2;
  n=kmax;

  abd=dvector(0,(jmax2+1)*(n+1));

```

```

/*zero the matrix*/
for(j=1;j<=n;j++)
{
  for(i=1;i<=jmax2;i++)
    abd[i*n+j]=0;
}

/*write in the non zero diagonals in band format*/
for(i=1;i<=n;i++)
  abd[jmax2*n+i]=d1C[i];
for(i=2;i<=n;i++)
  abd[m*n+i]=dbC[i-1];
for(i=jmax2;i<=n;i++)
  abd[l*n+i]=dcC[i-m];

/*do the L D L factoring*/

for(j=1;j<=n;j++)
{
  s=0;
  ik=m+1;
  jk=max0C(j-m,1);
  mu=max0C(m+2-j,1);
  if(m>=mu)
  {
    for(k=mu;k<=m;k++)
    {
      t=0;
      t=abd[k*n+j]-ddotC(k-mu,ik,jk,n,mu,j);
      t=t/abd[(m+1)*n+jk];
      abd[k*n+j]=t;
      s=s+(t*t);
      ik--;
      jk++;
    }
  }
  /*aquí significa que m fue menor a mu */
  s=abd[(m+1)*n+j]-s;
  abd[(m+1)*n+j]=sqrt(s);
}/*j*/

/*solve trans(R)*y=b*/
for(k=1;k<=n;k++)
{
  lm=min0C(k-1,m);
  la=m+1-lm;
  lb=k-lm;
  t=ddot2C(lm,la,k,n,&b[lb]);
  b[k]=(b[k]-t)/abd[(m+1)*n+k];
}

/*solve r*x=y*/

for(kb=1;kb<=n;kb++)
{
  k=n+1-kb;
  lm=min0C(k-1,m);
  la=m+1-lm;
  lb=k-lm;
  b[k]=b[k]/abd[(m+1)*n+k];
  t=b[k]*-1;
  daxpyC(lm,t,la,k,n,&b[lb]);
}
free((char *)abd);
}

void malla(int I1,int J1,double F1,double F2,double PHI1,double RH1,double R1)
{
  int i,j,k;
  double D1,D2;

```

```

R=dvector(0,J1);
PHIT1=dvector(0,I1);
PHIT2=dvector(0,I1);
jpot=ivector(0,5);
    if (F1 == 1.0)
        D1=PHI1/(I1-1);
    else
        D1=PHI1*(1-F1)/(1-pow(F1,I1));

    if (F2 == 1.0)
        D2=RH1/(J1-1);
    else
        D2=RH1*(1-F2)/(1-pow(F2,J1));

R[0]=R1;
for(j=1;j<J1;j++)
    {
        R[j]=R[j-1]+D2;
        D2=F2*D2;
    }

PHIT1[0]=1;
for(i=1;i<I1;i++)
    {
        PHIT1[i]=PHIT1[i-1]+D1;
        D1=F1*D1;
    }
PHIT1[I1-1]=PHI1-1;

for (k=0;k<4;k++)
    {
        for(j=0;j<J1;j++)
            {
                if (R[j]<tool[k] && R[j+1]>tool[k])
                    { R[j+1]=tool[k];
                      jpot[k]=j+1;
                    }
                R[J1-1]=RH1;
            }
    }

PHIT2[0]=PHIT1[0];
for (i=1;i<=I1;i++)
    {
        PHIT2[i]=(PHIT1[i]+PHIT1[i-1])/2;
        PHIT2[I1]=179;
    }

y=dvector(0,I1*J1);
for(i=0;i<I1;i++)
for(j=0;j<J1;j++)
    {
        y[i+1+j]=R[j]*cos(PHIT1[i]*P3/180);
    }
}

void xyohr(int I1,int J1,int nlay,float sourceprof)
{
    int i,j,k;

    Z=dvector(0,nlay);
    OHR=dvector(0,I1*J1);
    for(k=0;k<nlay-1;k++)
    {

```

```

    Z[k]=sourceprof-lay[k];
}

for(k=0;k<nlay-1;k++)
{ if (Z[k]>=-tool[0] && Z[k]<=0)
    Z[k]=0.0;
  for(i=0;i<=I1;i++)
  {
    for(j=0;j<=J1;j++)
    {

      if (y[i+I1*j]>Z[0])
        OHR[i+I1*j]=OHR[0];
      if (Z[k]<0)
        if (y[i+I1*j]<=Z[k] && y[i+I1*j]>Z[k+1])
          OHR[i+I1*j]=OHR[k];
        if (y[i+I1*j]<=Z[k] && y[i+I1*j]>Z[k+1])
          OHR[i+I1*j]=OHR[k+1];
        if (y[i+I1*j]<Z[nlay-2])
          OHR[i+I1*j]=OHR[nlay-1];

    }
  }
}

}

void nrerror(char error_text[])
/* Numerical Recipes standard error handler */
{
    fprintf(stderr,"Numerical Recipes run-time error...\n");
    fprintf(stderr,"%s\n",error_text);
    fprintf(stderr,"...now exiting to system...\n");
    exit(1);
}

int *ivector(long nl, long nh)
/* allocate an int vector with subscript range v[nl..nh] */
{
    int *v;

    v=(int *)malloc((size_t)((nh-nl+1+NR_END)*sizeof(int)));
    if (!v) nrerror("allocation failure in ivector()");
    return v-nl+NR_END;
}

double *dvector(long nl, long nh)
/* allocate a double vector with subscript range v[nl..nh] */
{
    double *v;

    v=(double *)malloc((size_t)((nh-nl+1+NR_END)*sizeof(double)));
    if (!v) nrerror("allocation failure in dvector()");
    return v-nl+NR_END;
}

double **dmatrix(long nrl, long nrh, long ncl, long nch)
/* allocate a double matrix with subscript range m[nrl..nrh][ncl..nch] */
{
    long i, nrow=nrh-nrl+1, ncol=nch-ncl+1;
    double **m;

    /* allocate pointers to rows */
    m=(double **) malloc((size_t)(nrow+NR_END)*sizeof(double*));
    if (!m) nrerror("allocation failure 1 in matrix()");
    m += NR_END;
    m -= nrl;

    /* allocate rows and set pointers to them */
    m[nrl]=(double *) malloc((size_t)(nrow*ncol+NR_END)*sizeof(double));
    if (!m[nrl]) nrerror("allocation failure 2 in matrix()");

```

```

    m[nrl] += NR_END;
    m[nrl] -= ncl;

    for(i=nrl+1; i<=nrh; i++) m[i]=m[i-1]+ncol;

    /* return pointer to array of pointers to rows */
    return m;
}

void free_ivector(int *v, long nl)
/* free an int vector allocated with ivector() */
{
    free((FREE_ARG) (v+nl-NR_END));
}

void free_dvector(double *v, long nl)
/* free a double vector allocated with dvector() */
{
    free((FREE_ARG) (v+nl-NR_END));
}

void free_dmatrix(double **m, long nrl, long ncl)
/* free a double matrix allocated by dmatrix() */
{
    free((FREE_ARG) (m[nrl]+ncl-NR_END));
    free((FREE_ARG) (m+nrl-NR_END));
}

main()
{

int I1,J1,i,j,n,zj,ni,l,k,m,t,ip,pi,N,aa,bb,nlay,kk,nmues;
float S9,L9,RH1,R1,PHI1,PHI2,PHI3,F1,F2,D1,D2,A1,A2,A3,A4,OHR0,R6,R3,TANPHI,M1,C3,D3;
float z,w,v,y,sumaa,suma,sum,smax,r1,c,sumaaa,dphi,dr;
float o,p,q,r,analytical,error,xx,yyy,V,fro,res,end,samp;
float prof,sourceprof,start,sn,ln,lat1,lat2,lat;

/* start to set geometry and grid */
printf("PROGRAMA DE DIFERENCIAS FINITAS ESFERICO\n");
printf("TESIS DE INGENIERO GEOFISICO\n");
printf("RICARDO ZAVALA TORRES:\n");
char name[14];
FILE *fp;
printf("Archivo a salvar:");
gets(name);
if (name[0]=='\0'){
    fprintf(stderr,"no filename typed\n");
    exit(EXIT_SUCCESS);
}
fp=fopen(name,"w");
if (fp==NULL){
    fprintf(stderr,"cannot open %s\n",name);
    exit(EXIT_FAILURE);
}

printf("number of nodes in r direction:\n");
scanf("%d",&J1);
printf("number of nodes in phi direction:\n");
scanf("%d",&I1);
fprintf(fp,"number of nodes in r direction: %d\n",J1);
fprintf(fp,"number of nodes in phi direction: %d\n",I1);

/* RH1 is the radius of dona and PHI1 is the angle of calculation zone */

```

```

/* internal radius of dona */
R1=0;

printf("internal radius of dona:\n");
scanf("%f",&R1);
printf("radial thickness of dona:\n");
scanf("%f",&RH1);
printf("angular zone of dona:\n");
scanf("%f",&PHI2);
fprintf(fp,"internal radius of dona: %f\n",R1);
fprintf(fp,"radial thickness of dona: %f\n",RH1);
fprintf(fp,"angular zone of dona: %f\n",PHI2);

/* *P3/180; */

/* F1 and F2 expansion factors in r and phi direction */

printf("Enter Grid-expansion factor in radial:\n");
scanf("%f",&F2);
printf("Enter Grid-expansion factor in Phi:\n");
scanf("%f",&F1);
fprintf(fp,"Enter Grid-expansion factor in radial: %f\n",F2);
fprintf(fp,"Enter Grid-expansion factor in Phi: %f\n",F1);
tool=dvector(0,5);
/* I1=14;
J1=20;
R1=0.01;
RH1=50;
PHI2=180;
F1=1;
F2=1.26;*/

PHI1=PHI2;
tool[0]=0.4064;
tool[1]=1.6256;
tool[2]=5.2832;
tool[3]=6.0960;
printf("Enter number of layers:\n");
scanf("%d",&nlay);
fprintf(fp,"Enter number of layers: %d\n",nlay);

/*nlay=3;*/
lay=dvector(0,nlay);
OHRS=dvector(0,nlay);

printf("start:\n");
scanf("%f",&start);
fprintf(fp,"start: %f\n",start);
printf("end:\n");
scanf("%f",&end);
fprintf(fp,"end: %f\n",end);
printf("samp rate:\n");
scanf("%f",&samp);
fprintf(fp,"samp rate: %f\n",samp);

/*lay[0]=1010;
lay[1]=1020;
lay[2]=10060;

OHRS[0]=0.5;
OHRS[1]=0.02;
OHRS[2]=0.5;*/

/*start=1000;*/
/*sourceprof=1000;*/
/* printf("prof=");

```

```

scanf("%f",&sourceprof);*/
for(i=0;i<nlay;i++)
{
    printf("Enter resistivity of layer %d:\n",i+1);
    scanf("%f",&res);
    fprintf(fp,"resistivity of layer %d: %f\n",i+1,res);
    OHRs[i]=1/res;
}
for(k=0;k<nlay-1;k++)
{
    printf("Enter depth of boundary %d:\n",k+1);
    scanf("%f",&fro);
    fprintf(fp,"depth of boundary %d: %f\n",k+1,fro);
    lay[k]=fro;
}
malla(I1,J1,F1,F2,PHI1,RH1,R1);
nmues=((end-start)/samp);
for(kk=0;kk<=nmues;kk++)
{
    sourceprof=start+(kk*samp);
    xyohr(I1,J1,nlay,sourceprof);
    W=dvector(0,I1*J1);
    S=dvector(0,I1*J1);
    /* for (i=0;i<I1;i++)
    {
        for (j=0;j<J1;j++)
        {

            printf("%4.2f",OHR[i][j]);
            fprintf(fp,"%4.2f",OHR[i][j]);
        }
        printf("\n");
        fprintf(fp,"\n");
    }
    */
for (i=0;i<I1;i++)
{
    for (j=0;j<J1;j++)
    {
        W[i+11*j]=(2*(OHR[i+11*j]*OHR[i+11*(j+1)]))/(OHR[i+11*j]+OHR[i+11*(j+1)]);
        S[i+11*j]=(2*(OHR[i+11*j]*OHR[(i+1)+11*j]))/(OHR[i+11*j]+OHR[(i+1)+11*j]);
    }
}
ni=(J1-1)*I1;

d1C=dvector(0,ni+1);
dcC=dvector(0,ni+1);
dbC=dvector(0,ni+1);

for (i=0;i<I1-1;i++)
{
    for(j=0;j<J1-1;j++)
    {

d1C[(i+j)+(I1-1)*j+1]=0.0;
dcC[(i+j)+(I1-1)*j+1]=0.0;
dbC[(i+j)+(I1-1)*j+1]=0.0;

```

```

}
}

for(i=1;i<I1-1;i++)
{
    o=(PHIT1[i]*P3/180);
    p=(PHIT1[i+1]*P3/180);
    q=(PHIT1[i-1]*P3/180);

    y=pow((R[0]+(R[1]-R[0])/2),2);
    yy=pow(R[0],2)*pow(R[1]-R[0],2);

    z=pow(R[0],2)*sin(o)*(p-o)*(p-q);
    zz=pow(R[0],2)*sin(o)*(o-q)*(p-q);

j=0;
V=2*P3*pow(R[0],2)*sin(o)*((R[1]-R[0])/2)*((p-q)/2);
d1C[(i+j)+(I1-1)*j+1]=+(V*(W[i+1]*j)*2*y)/yy
    +(V*S[i+1]*j)*2*sin(o+(p-o)/2)/z
    +(V*S[(i-1)+1]*j)*2*sin(o-(o-q)/2)/zz);

dcC[(i+j)+(I1-1)*j+1]=-V*(W[i+1]*j)*2*y)/yy;

dbC[(i+j)+(I1-1)*j+1]=-V*S[i+1]*j)*2*sin(o+(p-o)/2)/z;

/*
printf("%d,%10.5f,%10.5f,%10.5f,%10.5f\n", (i+j)+(I1-1)*j, V[(i+j)+(I1-1)*j], A[(i+j)+(I1-1)*j], B[(i+j)+(I1-1)*j], C[(i+j)+(I1-1)*j]);
fprintf(fp, "%d,%10.5f,%10.5f,%10.5f,%10.5f\n", (i+j)+(I1-1)*j, V[(i+j)+(I1-1)*j], A[(i+j)+(I1-1)*j], B[(i+j)+(I1-1)*j], C[(i+j)+(I1-1)*j]);
*/
}

/* general */
for (i=1;i<I1-1;i++)
{
    for(j=1;j<J1-1;j++)
    {

        o=(PHIT1[i]*P3/180);
        p=(PHIT1[i+1]*P3/180);
        q=(PHIT1[i-1]*P3/180);
        y=2*pow(R[j],3)*(R[j+1]-R[j-1]);

        yy=(2*R[j+1]*pow(R[j],2)+(R[j]*pow(R[j+1],2))+pow(R[j],3);
        yyy=(2*R[j-1]*pow(R[j],2)+(R[j]*pow(R[j-1],2))+pow(R[j],3);

        z=pow(R[j],2)*sin(o)*(p-o)*(p-q);
        zz=pow(R[j],2)*sin(o)*(o-q)*(p-q);

V=2*P3*pow(R[j],2)*sin(o)*((R[j+1]-R[j-1])/2)*((p-q)/2);
d1C[(i+j)+(I1-1)*j+1]=+(V*W[i+1]*j)*yy/(y*(R[j+1]-R[j]))
    +(V*W[i+1]*j-1)*yyy/(y*(R[j]-R[j-1]))
    +(V*S[i+1]*j)*2*sin(o+(p-o)/2)/z
    +(V*S[(i-1)+1]*j)*2*sin(o-(o-q)/2)/zz);

dcC[(i+j)+(I1-1)*j+1]=-V*W[i+1]*j)*yy/(y*(R[j+1]-R[j]));

dbC[(i+j)+(I1-1)*j+1]=-V*S[i+1]*j)*2*sin(o+(p-o)/2)/z;

/*
printf("%d,%10.5f,%10.5f,%10.5f,%10.5f\n", (i+j)+(I1-1)*j, V[(i+j)+(I1-1)*j], A[(i+j)+(I1-1)*j], B[(i+j)+(I1-1)*j], C[(i+j)+(I1-1)*j]);
fprintf(fp, "%d,%10.5f,%10.5f,%10.5f,%10.5f\n", (i+j)+(I1-1)*j, V[(i+j)+(I1-1)*j], A[(i+j)+(I1-1)*j], B[(i+j)+(I1-1)*j], C[(i+j)+(I1-1)*j]);
*/
}
}

```

```

}
}
o=0.0;
p=0.0;
q=0.0;
y=0.0;
yy=0.0;
z=0.0;
zz=0.0;
/* frontera izquierda */

for (j=1;j<J1-1;j++)
{

o=(PHIT1[0]*P3/180);
p=(PHIT1[1]*P3/180);

y=2*pow(R[j],3)*(R[j+1]-R[j-1]);

yy=(2*R[j+1]*pow(R[j],2)+(R[j]*pow(R[j+1],2))+pow(R[j],3));
yyy=(2*R[j-1]*pow(R[j],2)+(R[j]*pow(R[j-1],2))+pow(R[j],3));
z=pow(R[j],2)*sin(o)*pow(p-o,2);

i=0;
V=2*P3*pow(R[j],2)*sin(o)*((R[j+1]-R[j-1])/2)*((p-o)/2);

d1C[(i+j)+(I1-1)*j+1]=(V*W[i+1]*j)*yy/(y*(R[j+1]-R[j]))
+(V*W[i+1]*(j-1))*yyy/(y*(R[j]-R[j-1]))
+(V*S[i+1]*j)*2*sin(o+(p-o)/2)/z;

dcC[(i+j)+(I1-1)*j+1]=-V*W[i+1]*j*yy/(y*(R[j+1]-R[j]));

dbC[(i+j)+(I1-1)*j+1]=-V*S[i+1]*j*2*sin(o+(p-o)/2)/z;
/* C[(i+j)+(I1-1)*j]=0.0;*/
/*
printf("%d,%10.5f,%10.5f,%10.5f,%10.5f\n", (i+j)+(I1-1)*j, V[(i+j)+(I1-1)*j], A[(i+j)+(I1-1)*j], B[(i+j)+(I1-1)*j], C[(i+j)+(I1-1)*j]);
fprintf(fp, "%d,%10.5f,%10.5f,%10.5f,%10.5f\n", (i+j)+(I1-1)*j, V[(i+j)+(I1-1)*j], A[(i+j)+(I1-1)*j], B[(i+j)+(I1-1)*j], C[(i+j)+(I1-1)*j]);
*/

}

/* frontera derecha */

o=0.0;
p=0.0;
q=0.0;
y=0.0;
yy=0.0;
z=0.0;
zz=0.0;

for (j=1;j<J1-1;j++)
{

p=(PHIT1[I1-1]*P3/180);
o=(PHIT1[I1-2]*P3/180);

y=2*pow(R[j],3)*(R[j+1]-R[j-1]);

yy=(2*R[j+1]*pow(R[j],2)+(R[j]*pow(R[j+1],2))+pow(R[j],3));
yyy=(2*R[j-1]*pow(R[j],2)+(R[j]*pow(R[j-1],2))+pow(R[j],3));
z=pow(R[j],2)*sin(p)*pow(p-o,2);

```

```

i=i-1;
V=2*P3*pow(R[j],2)*sin(p)*((R[j+1]-R[j-1])/2)*((p-o)/2);
d1C[(i+j)+(1-1)*j+1]=+(V*W[i+1]*j)*yy/(y*(R[j+1]-R[j]))
                    +(V*W[i+1]*(j-1))*yyy/(y*(R[j]-R[j-1]))
                    +(V*S[(i-1)+1]*j)*2*sin(p-(p-o)/2)/z);

dcC[(i+j)+(1-1)*j+1]=V*W[i+1]*j)*yy/(y*(R[j+1]-R[j]));
dbC[(i+j)+(1-1)*j+1]=0.0;

/*
printf("%d,%10.5f,%10.5f,%10.5f,%10.5f\n", (i+j)+(1-1)*j, V[(i+j)+(1-1)*j], A[(i+j)+(1-1)*j], B[(i+j)+(1-1)*j], C[(i+j)+(1-1)*j]);
fprintf(fp, "%d,%10.5f,%10.5f,%10.5f,%10.5f\n", (i+j)+(1-1)*j, V[(i+j)+(1-1)*j], A[(i+j)+(1-1)*j], B[(i+j)+(1-1)*j], C[(i+j)+(1-1)*j]);
*/
}

```

```

o=(PHIT1[0]*P3/180);
p=(PHIT1[1]*P3/180);
y=pow((R[0]+(R[1]-R[0])/2),2);
yy=pow(R[0],2)*pow(R[1]-R[0],2);

```

```
z=pow(R[0],2)*sin(o)*pow(p-o,2);
```

```

i=0;
j=0;
V=2*P3*pow(R[0],2)*sin(o)*((R[1]-R[0])/2)*((p-o)/2);

dcC[(i+j)+(1-1)*j+1]=V*(W[i+1]*j)*2*y)/yy;
dbC[(i+j)+(1-1)*j+1]=V*S[i+1]*j)*2*sin(o+(p-o)/2)/z;

d1C[(i+j)+(1-1)*j+1]=+(V*(W[i+1]*j)*2*y)/yy
                    +(V*S[i+1]*j)*2*sin(o+(p-o)/2)/z;

```

```

o=(PHIT1[1-2]*P3/180);
p=(PHIT1[1-1]*P3/180);
y=pow((R[0]+(R[1]-R[0])/2),2);
yy=pow(R[0],2)*pow(R[1]-R[0],2);
z=pow(R[0],2)*sin(p)*pow(p-o,2);

```

```

j=0;
i=i-1;
V=2*P3*pow(R[0],2)*sin(p)*((R[1]-R[0])/2)*((p-o)/2);
d1C[(i+j)+(1-1)*j+1]=+(V*(W[i+1]*j)*2*y)/yy
                    +(V*S[(i-1)+1]*j)*2*sin(p-(p-o)/2)/z);

```

```
dcC[(i+j)+(1-1)*j+1]=-(V*W[i+1]*j)*2*y)/yy;
```

```
dbC[(i+j)+(1-1)*j+1]=0.0;
```

```

/*
for(i=0;i<ni;i++)
{
    for(j=0;j<ni;j++)
    {
        a[i*ni+j]=0.0;
        if(i==j)a[i*ni+j]=-A[i]*V[i];
        if((i+j)==1+(j*2))a[i*ni+j]=-B[i]*V[i];
        if((i+j)==1+(i*2))a[i*ni+j]=-C[i]*V[i];
        if((i+j)==1+(j*2))a[i*ni+j]=-D[j]*V[i];
        if((i+j)==1+(i*2))a[i*ni+j]=-E[j]*V[i];
        printf("%10.5f",a[i*ni+j]);
        fprintf(fp, "%10.5f",a[i*ni+j]);
    }
}

```

```

printf("\n");
fprintf(fp, "\n");
} */
/*      for(j=0;j<ni;j++)
    {

        d1C[j+1]=0.0;
        dcC[j+1]=0.0;
        dbC[j+1]=0.0;

    }

    for(i=0;i<ni;i++)
    {
        for(j=0;j<ni;j++)
        {

            if(i==j) d1C[i+1]=-A[i];
            if((i+j)==1+(i*2)) dcC[i+1]=-B[i];
            if((i+j)==1+(i*2)) dbC[i+1]=-C[i];

        }

    } */
for(k=0;k<J1-1;k++)
for(i=0;i<ni;i++)
{

    if(j==(k*(11))+(11-1))
    dbC[i+1]=0;

}
b=dvector(0,ni+1);
for (j=1;j<=ni;j++)

{

    b[j]=0.0;
    if(j==(11/2)) b[j]=1;

/* printf("%d,%10.5f,%10.5f,%10.5f,%10.5f\n",j,b[j],d1C[j],dbC[j],dcC[j]);
fprintf(fp, "%d,%10.5f,%10.5f,%10.5f,%10.5f\n",j,b[j],d1C[j],dbC[j],dcC[j]); */

}

/*ni-=1;*/
N=ni;
adaptC(11-1,N);

        for (i=0;i<N;i++){

            b[i]=b[i+1];
            /*
            printf("%d,%10.5f\n",i,b[i]);
            fprintf(fp, "%d,%10.5f\n",i,b[i]);
            */
        }

/*fprintf(fp, "\n");
printf("\n");*/
/*
for (i=0;i<11;i++)
{
    for (j=0;j<11-1;j++)
    {

```

```

        xx=R[j]*sin(PHIT1[i]*P3/180);
        y=R[j]*cos(PHIT1[i]*P3/180);
        z=b[i+j+((11-1)*j)];
        zz=d1C[i+j+((11-1)*j)+1];
        analytical=dbC[i+j+((11-1)*j)+1];
        error=dcC[i+j+((11-1)*j)+1];

printf("%d %d %6.3f %8.6f %8.6f %6.4f %8.6f %8.6f\n",i,j,xx,y,z,zz,analytical,error );
fprintf(fp,"%d %d %6.3f %8.6f %8.6f %6.4f %8.6f %8.6f\n",i,j,xx,y,z,zz,analytical,error );
    }
}

printf("No x y pot rhoapp rad anal error\n");
fprintf(fp,"No x y pot rhoapp rad anal error\n");
for (j=0;j<J1;j++)
{ i=11-1;
  for (i=0;i<11;i++)
  {

    if (j==J1-1) b[i+j+((11-1)*j)]=0.0;
    xx=R[j]*sin(PHIT1[i]*P3/180);
    y=R[j]*cos(PHIT1[i]*P3/180);
    z=b[i+j+((11-1)*j)];
    zz=R[j]*4*P3*b[i+j+((11-1)*j)];
    analytical=(1/0.02)/(4*P3*R[j]);
    error=100-(z/analytical)*100);

    printf("%d %6.3f %6.4f %8.6f %8.6f %6.4f %8.6f %8.6f\n", (i+j)+(11-1)*j,xx,y,z,zz,R[j],analytical,error );
    fprintf(fp,"%d %6.3f %6.4f %8.6f %8.6f %6.4f %8.6f %8.6f\n", (i+j)+(11-1)*j,xx,y,z,zz,R[j],analytical,error );
  }
}
*/
i=11-1;
j=jpot[0];
sn=R[j]*4*P3*b[i+j+((11-1)*j)]*1.08379;
/* j=jpot[1];
ln=R[j]*4*P3*b[i+j+((11-1)*j)];
j=jpot[3];
lat1=b[i+j+((11-1)*j)];
j=jpot[2];
lat2=b[i+j+((11-1)*j)];
lat=497.92987*(lat2-lat1);
printf("%6.4f %8.6f %8.6f %8.6f\n",sourceprof,sn,ln,lat);
fprintf(fp,"%6.4f %8.6f %8.6f %8.6f\n",sourceprof,sn,ln,lat);
*/
printf("%6.4f %8.6f\n",sourceprof,sn);
fprintf(fp,"%6.4f %8.6f\n",sourceprof,sn);

free_dvector(OHR,0);
free_dvector(W,0);
free_dvector(S,0);

free_dvector(b,0);

free_dvector(dbC,0);
free_dvector(dcC,0);
free_dvector(d1C,0);
free_dvector(Z,0);
}
free_dvector(OHRS,0);
free_dvector(R,0);
free_dvector(PHIT1,0);
fclose(fp);
return EXIT_SUCCESS;
}

```

Programa de resistencias eléctricas cilíndrico

```

/*Programa de Resistencias eléctricas cilindrico
 * Ricardo Zavala Torres
 */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#include "nrutil.h"
#define NR_END 0
#define FREE_ARG char*
#define P3 atan(1)*4

double *tool;
double *R2;
double *PHI;
double **OHR;
double **S;
double **W;
double yy,zz;
double *abd;
double *d1C;
double *dbC;
double *dcC;
double *b,*x;
double *lay;
double *Z1;
double *profabs;
double *OHR1;
int *OHR2;
int *tooli;

void nerror(char error_text[])
/* Numerical Recipes standard error handler */
{
    fprintf(stderr,"Numerical Recipes run-time error...\n");
    fprintf(stderr,"%s\n",error_text);
    fprintf(stderr,"...now exiting to system...\n");
    exit(1);
}

int *ivector(long nl, long nh)
/* allocate an int vector with subscript range v[nl..nh] */
{
    int *v;

    v=(int *)malloc((size_t) ((nh-nl+1+NR_END)*sizeof(int)));
    if (!v) nerror("allocation failure in ivector()");
    return v-nl+NR_END;
}

double *dvector(long nl, long nh)
/* allocate a double vector with subscript range v[nl..nh] */
{
    double *v;

    v=(double *)malloc((size_t) ((nh-nl+1+NR_END)*sizeof(double)));
    if (!v) nerror("allocation failure in dvector()");
    return v-nl+NR_END;
}

double **dmatrix(long nrl, long nrh, long ncl, long nch)
/* allocate a double matrix with subscript range m[nrl..nrh][ncl..nch] */
{
    long i, nrow=nrh-nrl+1, ncol=nch-ncl+1;
    double **m;

    /* allocate pointers to rows */
    m=(double **) malloc((size_t)(nrow+NR_END)*sizeof(double**));
    if (!m) nerror("allocation failure 1 in matrix()");
    m += NR_END;
    m -= nrl;
}

```

```

/* allocate rows and set pointers to them */
m[nr1]=(double *) malloc((size_t)((nrow*ncol+NR_END)*sizeof(double)));
if (!m[nr1]) perror("allocation failure 2 in matrix()");
m[nr1] += NR_END;
m[nr1] -= ncl;

for(i=nr1+1;i<=nrh;i++) m[i]=m[i-1]+ncol;

/* return pointer to array of pointers to rows */
return m;
}

void free_ivector(int *v, long nl)
/* free an int vector allocated with ivector() */
{
    free((FREE_ARG) (v+nl-NR_END));
}

void free_dvector(double *v, long nl)
/* free a double vector allocated with dvector() */
{
    free((FREE_ARG) (v+nl-NR_END));
}

void free_dmatrix(double **m, long nr1, long ncl)
/* free a double matrix allocated by dmatrix() */
{
    free((FREE_ARG) (m[nr1]+ncl-NR_END));
    free((FREE_ARG) (m+nr1-NR_END));
}

int max0C(int num1,int num2)
{
    int regreso;
    if(num1>=num2)
        regreso=num1;
    else
        regreso=num2;

    return(regreso);
}

int min0C(int num1,int num2)
{
    int regreso;
    if(num1<=num2)
        regreso=num1;
    else
        regreso=num2;

    return(regreso);
}

double ddotC(int n, int la,int k,int N,int mu,int j)
{
    int i;
    double regreso;

    regreso=0;
    for(i=0;i<n;i++)
        regreso+=abd[(la+i)*N+k]*abd[(mu+i)*N+j];
    return(regreso);
}

double ddot2C(int n,int la, int k, int N,double *dy)
{
    int i;
    double regreso;

    regreso=0;
    for(i=0;i<n;i++)

```

```

{
    regreso+=abd[(la+i)*N+k]*(*dy);
    dy++;
}
return(regreso);
}

void daxpyC(int n,double da,int la,int k,int N, double *dy)
{
    int m,i,mp1;
    double *dyanterior;

    dyanterior=dy;

    m=n%4;
    for(i=1;i<=m;i++)
    {
        *dy=*dy+da*abd[(la+i-1)*N+k];
        dy++;
    }

    dy=dyanterior;
    if(n>=4)
    {
        mp1=m+1;
        dy+=m;
        for(i=mp1;i<=n;i++)
        {
            *dy=*dy+da*abd[(la+i-1)*N+k];
            dy++;
        }
    }
}

void adaptC(int jmax,int kmax)
{
    int i,k0,j,k,ik,jk,mu;
    int la,lm,lb,kb;
    int m,n,jmax2;
    double s,t;
    FILE *arch;

    /* write the b vector to be solved*/
    /* b=(double *)propio_malloc((kmax+1)*sizeof(double));

    for(i=1;i<=kmax;i++)
        b[i]=0;
    k0=(jmax+1)*(i0-1)+1;
    b[k0]=current;*/

    /*Band Storage*/
    m=jmax+1;
    jmax2=jmax+2;
    n=kmax;

    abd=dvector(0,(jmax2+1)*(n+1));

    /* zero the matrix*/
    for(j=1;j<=n;j++)
    {
        for(i=1;i<=jmax2;i++)
            abd[i*n+j]=0;
    }

    /*write in the non zero diagonals in band format*/
    for(i=1;i<=n;i++)
        abd[jmax2*n+i]=d1C[i];
    for(i=2;i<=n;i++)
        abd[m*n+i]=dbC[i-1];
}

```

```

for(i=jmax2;j<=n;i++)
abd[1*n+i]=dcC[i-m];

/*do the L D L factoring*/

for(j=1;j<=n;j++)
{
s=0;
ik=m+1;
jk=max0C(j-m,1);
mu=max0C(m+2-j,1);
if(m>=mu)
{
for(k=mu;k<=m;k++)
{
t=0;
t=abd[k*n+j]-ddotC(k-mu,ik,jk,n,mu,j);
t=t/abd[(m+1)*n+jk];
abd[k*n+j]=t;
s=s+(t*t);
ik--;
jk++;
}
}
/*aqui significa que m fue menor a mu */
s=abd[(m+1)*n+j]-s;
abd[(m+1)*n+j]=sqrt(s);
}/**/

/*solve trans(R)*y=b*/
for(k=1;k<=n;k++)
{
lm=min0C(k-1,m);
la=m+1-lm;
lb=k-lm;
t=ddot2C(lm,la,k,n,&b[lb]);
b[k]=(b[k]-t)/abd[(m+1)*n+k];
}

/*solve r*x=y*/

for(kb=1;kb<=n;kb++)
{
k=n+1-kb;
lm=min0C(k-1,m);
la=m+1-lm;
lb=k-lm;
b[k]=b[k]/abd[(m+1)*n+k];
t=b[k]*-1;
daxpyC(lm,t,la,k,n,&b[lb]);
}
free_dvector(abd,0);
}

void malla(int I1,int J1,double F1,double F2,double PHI1,double RH1,double R1)
{

int i,j,k;
double D1,D11,D2,zz;

R2=dvector(0,J1);
PHI=dvector(0,I1);
tooli=ivector(0,5);
Z1=dvector(0,I1);

if (F1 == 1.0)

D1=PHI1/(I1-1);
else

```

```

        {
            D1=(PHI1)*(1-F1)/(1-pow(F1,J1));
            D11=(PHI1)*(1-F1)/(1-pow(F1,J1));
        }
if (F2 == 1.0)
    D2=RH1/(J1-1);
else
    D2=RH1*(1-F2)/(1-pow(F2,J1));

R2[0]=R1;
/*printf("%d,%6.2fn",0,R2[0]);*/
for(j=1;j<J1;j++)
    {
        R2[j]=R2[j-1]+D2;
        D2=F2*D2;

        /* printf("%d,%6.2fn",j,R2[j]); */
    }
/*for(j=0;j<J1;j++)
    {
        if (fabs(R2[j]- tool)<0.07 ) R2[j]=tool;
        printf("%d,%6.4fn",j,R2[j]);
    }
*/

/* for(j=0;j<J1;j++)
    {
        if (R2[j]<tool) R2[j+1]=tool;
        R2[J1-1]=R2H1;
        printf("%d,%6.4fn",j,R2[j]);
        fprintf(fp,"%d,%6.4fn",j,R2[j]);
    }*/

for (k=0;k<1;k++)
for(j=0;j<J1;j++)
    {
        if (R2[j]<tool[k] && R2[j+1]>tool[k] ) R2[j+1]=tool[k];

        /* printf("%d,%6.4fn",j,R2[j]);
        fprintf(fp,"%d,%6.4fn",j,R2[j]);*/
    }
R2[J1-1]=RH1;
/*PHI[0]=0;
printf("%d,%6.2fn",0,PHI[0]);
for(i=1;i<11;i++)
    {
        PHI[i]=PHI[i-1]+D1;
        D1=F1*D1;
        printf("%d,%6.2fn",i,PHI[i]);
    }*/

PHI[0]=0.0;
PHI[(11-1)/2]=PHI1/2;
for(i=(11-1)/2-1;i>0;i--)
    {
        PHI[i]=PHI[i+1]-D1;
        D1=F1*D1;
        /*printf("%d,%6.2f,%6.2fn",i,PHI[i],D1);
        fprintf(fp,"%d,%6.2f,%6.2fn",i,PHI[i],D1);*/
    }

for(i=(11-1)/2+1;i<11;i++)
    {

```

```

        PHI[i]=PHI[i-1]+D11;
        D11=F1*D11;

    }
    PHI[I1-1]=PHI1;
    for (k=0;k<1;k++)
    for (j=(I1-1)/2+1;j<I1;j++)
    {
        zz=(PHI1/2+tool[k]);
        if (PHI[i]<zz && PHI[i+1]>zz)
            { PHI[i+1]=(PHI1/2)+tool[k];
              tooli[k]=i+1;
            }
    }

}

for(i=0;i<I1;i++)
{
    Z1[i]=PHI1/2-PHI[i];
}

}
void xyrho1(int I1,int J1,int nlay,float sourceprof)
{
    int i,j,k;

    OHR=dmatrix(0,I1+1,0,J1+1);
    profabs=dvector(0,nlay+1);

    for (i=0;i<nlay-1;i++)
    {
        profabs[i]=(sourceprof-lay[i]);
    }

    for (k=0;k<nlay-1;k++)
    {
        for(j=0;j<=J1;j++)
            for(i=0;i<=I1;i++)
            {
                if (profabs[k]==Z1[i])
                    OHR[i][j]=OHR1[k];
                if (profabs[k]>Z1[i] && profabs[k+1]<Z1[i])
                    OHR[i][j]=OHR1[k+1];

                if (profabs[0]<Z1[i])
                    OHR[i][j]=OHR1[0];
                if (profabs[nlay-2]>Z1[i])
                    OHR[i][j]=OHR1[nlay-1];
            }
    }
}

}

double sr(int i,int j)
{
    float sr,z,y;
    z=(PHI[i]+PHI[i-1])/2;
    y=(PHI[i+1]+PHI[i])/2;

    sr=((OHR[i][j+1]+OHR[i][j])/(2*(OHR[i][j+1]*OHR[i][j])))
        *(log(R2[j+1]/R2[j]))
        /(2*P3*(y-z));
}

```

```

    return sr;
}
double pr(int i,int j)
{
    float o,q,pr;
    o=(PHI[i]);
    q=(PHI[i+1]);
    pr=((((OHR[i+1][j]+OHR[i][j])/(2*(OHR[i+1][j]*OHR[i][j])))
        *(q-o))
        /(P3*(pow((R2[j+1]+R2[j])/2,2)-pow((R2[j]+R2[j-1])/2,2))));
    return pr;
}

```

```

void matriz(int I1,int J1,int ni)
{ int i,j,k;
float o,p,q,y,yy,yyy,z,zz,V;

```

```

b=dvector(0,ni+1);
d1C=dvector(0,ni+1);
dcC=dvector(0,ni+1);
dbC=dvector(0,ni+1);
x=dvector(0,ni+1);

```

```

for (i=1;i<I1-1;i++)
{
    for(j=0;j<J1-1;j++)

```

```

{
    d1C[(i+j)+(J1-2)*(i-1)]=0.0;
    dcC[(i+j)+(J1-2)*(i-1)]=0.0;
    dbC[(i+j)+(J1-2)*(i-1)]=0.0;
}
}

```

```

for (i=1;i<I1-1;i++)
{
    j=0;
    d1C[(i+j)+(J1-2)*(i-1)]=1/sr(i,0)+1/pr(i,0)+1/pr(i-1,0);
    dbC[(i+j)+(J1-2)*(i-1)]=-1/sr(i,0);

    dcC[(i+j)+(J1-2)*(i-1)]=-1/pr(i,0);
    if(j==I1-2)
        dcC[(i+j)+(J1-2)*(i-1)]=0.0;
    /*printf("%d,%10.5f,%10.5f,%10.5fn",i,d1C[i],dbC[i],dcC[i]);*/
}

```

```

for (i=1;i<I1-1;i++)
{
    for(j=1;j<J1-1;j++)

```

```

{
    d1C[(i+j)+(J1-2)*(i-1)]=1/sr(i,j)+1/sr(i,j-1)+1/pr(i,j)+1/pr(i-1,j);
    dbC[(i+j)+(J1-2)*(i-1)]=-1/sr(i,j);
}
}

```

```

dcC[(i+j)+(J1-2)*(i-1))=-1/pr(i,j);
if(j==J1-2)
  dbC[(i+j)+(J1-2)*(i-1)]=0.0;
  if(i==11-2)
    dcC[(i+j)+(J1-2)*(i-1)]=0.0;

}
}

/* for(i=0;i<ni;i++)
{
  for(j=0;j<ni;j++)
  {
a[i*ni+j]=0;
if(i==j) a[i*ni+j]=A[i]*V[i];
if((i+j)==(11-2)+(i*2))a[i*ni+j]=-B[i]*V[i];
if((i+j)==1+(i*2))a[i*ni+j]=-C[i]*V[i];
if((i+j)==(11-2)+(j*2))a[i*ni+j]=-B[j]*V[j];
if((i+j)==1+(j*2))a[i*ni+j]=-C[j]*V[j];
    fprintf(fp,"%10.5f",a[i*ni+j]);
  }
  fprintf(fp,"\n");
}*/

for (i=1;i<=ni;i++)
{

  b[i]=0.0;
  x[i-1]=0;
  if(i==(11-1)/2+1+(J1-2)*((11-1)/2-1)) {b[i]=1;
    x[i-1]=1;}

/* printf("%3.2f\n",b[i]);
fprintf(fp,"%3.2f\n",b[i]);*/
/* fprintf(fp,"%d,%10.5f,%10.5f,%10.5f,%10.5f\n",i,b[i],d1C[i],dbC[i],dcC[i]);*/
/* printf("%d,%10.5f,%10.5f,%10.5f,%10.5f\n",i,b[i],d1C[i],dbC[i],dcC[i]);*/
}

}

void xyrho(int I1,int J1,int nlay,float PHI1,float sourceprof)
{ int i,j,k;
OHR=dmatrix(0,I1+1,0,J1+1);
Z1=dvector(0,nlay+1);
profabs=dvector(0,nlay+1);
OHR2=ivector(0,nlay+1);

for (i=0;i<nlay-1;i++)
{
  profabs[i]=(sourceprof-lay[i]);

  if (profabs[i]>0)

    Z1[i]=PHI1/2-fabs(profabs[i]);
  else
    Z1[i]=PHI1/2+fabs(profabs[i]);
}
}

```

```

for (k=0;k<nlay-1;k++)
{
for(i=0;i<=11;i++)
{

    if (PHI[i]<Z1[k] && PHI[i+1]>Z1[k])

    {
    PHI[i]=Z1[k];
    OHR2[k]=i;
    }
    else
    if (PHI[i]==Z1[k])
    {
        /*PHI[i]=Z1[k];*/
        OHR2[k]=i; }
    }
}
for (i=0;i<=11;i++)
    {
    for (j=0;j<=J1;j++)
    {
        OHR[i][j]=OHR1[nlay-1];
    }
}

for(k=0;k<nlay-1;k++)
{
    for (i=0;i<=OHR2[1-k];i++)
    {
        for (j=0;j<=J1;j++)
            OHR[i][j]=OHR1[1-k];
    }
}

}
main()
{

int I1,J1,i,j,n,zj,ni,l,k,m,t,ip,pi,N,aa,bb,nlay,nmues;
float RH1,R1,PHI1,PHI2,PHI3,F1,F2,D1,D11,D2,A1,A2,A3,A4,OHR0,R6,R3,TANPHI,M1,C3,D3;
float z,w,v,y,sumaa,suma,sum,smax,r1,c,sumaaa,dphi,dr;
float o,p,q,r,analytical,error,xx,yyy,start,sourceprof,kk,fro,res,end,samp;
/* start to set geometry and grid */

char name[14];
FILE *fp;
printf("Archivo a salvar:");
gets(name);
if (name[0]=='\0'){
fprintf(stderr, "no filename typed\n");
exit(EXIT_SUCCESS);
}
fp=fopen(name, "w+");
if (fp==NULL){
fprintf(stderr, "cannot open %s\n", name);
exit(EXIT_FAILURE);
}

tool=dvector(0,5);

```

```

printf("number of nodes in r direction:\n");
scanf("%d",&J1);
printf("number of nodes in phi direction:\n");
scanf("%d",&I1);
fprintf(fp,"number of nodes in r direction: %d\n",J1);
fprintf(fp,"number of nodes in z direction: %d\n",I1);

/* RH1 is the radius of dona and PHI1 is the angle of calculation zone */

/* internal radius of dona */

R1=0;

printf("internal radius of cylinder:\n");
scanf("%f",&R1);
printf("radial thickness of cylinder:\n");
scanf("%f",&RH1);
printf("length of cylinder:\n");
scanf("%f",&PHI2);
fprintf(fp,"internal radius of cylinder: %f\n",R1);
fprintf(fp,"radial thickness of cylinder: %f\n",RH1);
fprintf(fp,"length of cylinder: %f\n",PHI2);

/*PHI1=PHI2;*/
/* *P3/180; */

/* F1 and F2 expansion factors in r and phi direction */

printf("Enter Grid-expansion factor in radial:\n");
scanf("%f",&F2);
printf("Enter Grid-expansion factor in z:\n");
scanf("%f",&F1);
fprintf(fp,"Enter Grid-expansion factor in radial: %f\n",F2);
fprintf(fp,"Enter Grid-expansion factor in z: %f\n",F1);

/* J1=12;
I1=31;
R1=0.01;
RH1=50;
PHI2=100;
F1=1.2;
F2=1.8;*/
PHI1=PHI2;
ni=(J1-1)*(I1-2);
tool[0]=0.4064;
tool[1]=1.6256;
tool[2]=5.2832;
tool[3]=6.0960;
printf("Enter number of layers:\n");
scanf("%d",&nlay);
fprintf(fp,"Enter number of layers: %d\n",nlay);
/* nlay=3;*/
lay=dvector(0,nlay+1);
OHR1=dvector(0,nlay+1);

/*start=1000;*/
printf("start:\n");
scanf("%f",&start);
fprintf(fp,"start: %f\n",start);
printf("end:\n");
scanf("%f",&end);
fprintf(fp,"end: %f\n",end);
printf("samp rate:\n");
scanf("%f",&samp);
fprintf(fp,"samp rate: %f\n",samp);

```

```

/*
lay[0]=1010;
lay[1]=1020;
lay[2]=10060;
OHR1[0]=0.5;
OHR1[1]=0.02;
OHR1[2]=0.5;
OHR1[3]=0.05;
*/

for(i=0;i<nlay;i++)
{
printf("Enter resistivity of layer %d:\n",i+1);
scanf("%f",&res);
fprintf(fp,"resistivity of layer %d: %f\n",i+1,res);
OHR1[i]=1/res;
}
for(k=0;k<nlay-1;k++)
{
printf("Enter depth of boundary %d:\n",k+1);
scanf("%f",&fro);
fprintf(fp,"depth of boundary %d: %f\n",k+1,fro);
lay[k]=fro;
}
malla(I1,J1,F1,F2,PHI1,RH1,R1);

/* sourceprof=1009.250; */
/*printf("prof:");
scanf("%f",&sourceprof);*/
nmues=((end-start)/samp);
for(k=0;k<=nmues;k++)
{
sourceprof=start+(k*samp);

xyrho1(I1,J1,nlay,sourceprof);

/*xyrho(I1,J1,nlay,PHI1,sourceprof);*/

/*
W=dmatrix(0,I1,0,J1);
S=dmatrix(0,I1,0,J1);

for (i=0;i<I1;i++)
{
for (j=0;j<J1;j++)
{

W[i][j]=(2*(OHR[i][j]*OHR[i][j+1]))/(OHR[i][j]+OHR[i][j+1]);
S[i][j]=(2*(OHR[i][j]*OHR[i+1][j]))/(OHR[i][j]+OHR[i+1][j]);

}

} */

matriz(I1,J1,ni);
/*ni-=1;*/
N=ni;
adaptC(J1-2,N);
/*fprintf(fp," b, x, d1, d2, d3\n");*/
for (i=0;i<N;i++){
b[i]=b[i+1];

/*
fprintf(fp,"%d,%10.5f,%10.5f,%10.5f,%10.5f\n",i,x[i],b[i+1],d1C[i+1],dbC[i+1],dcC[i+1]);*/
}

```

```

/*fprintf(fp,"n");
printf("n");*/
/*
fprintf(fp,"i, j, z, r, x, d1, d2, d3, b\n");
for (i=0;i<I1-2;i++)
{
for (j=0;j<J1-1;j++)
{

xx=PHI[i+1];
y=R2[j];
z=b[i+j+((I1-3)*j)];

zz=d1C[i+j+((I1-3)*j)+1];
analytical=dbC[i+j+((I1-3)*j)+1];
error=dcC[i+j+((I1-3)*j)+1];

printf("%d %d %6.4f %8.6f %8.6f %6.4f%8.6f %8.6f %6.4f\n",i,j,xx,y,z,zz,analytical,error,x[i+j+((I1-3)*j)] );
fprintf(fp,"%d %d %6.4f %8.6f %8.6f %6.4f%8.6f %8.6f %6.4f\n",i,j,xx,y,z,zz,analytical,error,x[i+j+((I1-3)*j)] );
}
}

printf("No x y pot rhoapp rad anal error\n");
fprintf(fp,"No x y pot rhoapp rad anal error\n");*/
/*for (j=0;j<J1;j++)
{ for (i=0;i<I1-2;i++)
{
if (j==J1-1) b[i+j+((I1-3)*j)]=0.0;
xx=PHI[i+1];
y=R2[j];
z=b[i+j+((I1-3)*j)];
zz=4064*4*P3*b[i+j+(I1-3)*j];
analytical=(1/0.02)/(4*P3*R2[j]);
error=100-(100*z/analytical);

printf("%d %6.3f %6.4f %8.6f %8.6f %6.4f%8.6f %8.6f\n", (i+j)+(I1-3)*j,xx,y,z,zz,R2[j],analytical,error );
fprintf(fp,"%d %6.3f %6.4f %8.6f %8.6f %6.4f%8.6f %8.6f\n", (i+j)+(I1-3)*j,xx,y,z,zz,R2[j],analytical,error );
}
}*/
/* for (j=0;j<J1;j++)
{ i=tooli[0]-3;

if (j==J1-1) b[i+j+((I1-3)*j)]=0.0;
xx=PHI[i+1];
y=R2[j];
z=b[i+j+((I1-3)*j)];
zz=4064*4*P3*b[i+j+(I1-3)*j];
analytical=(1/0.02)/(4*P3*R2[j]);
error=100-(100*z/analytical);

printf("%d %6.3f %6.4f %8.6f %8.6f %6.4f%8.6f %8.6f\n", (i+j)+(I1-3)*j,xx,y,z,zz,R2[j],analytical,error );
fprintf(fp,"%d %6.3f %6.4f %8.6f %8.6f %6.4f%8.6f %8.6f\n", (i+j)+(I1-3)*j,xx,y,z,zz,R2[j],analytical,error );
}
}*/
i=tooli[0]-1;
j=0;
z=4064*4*P3*b[i+j+(J1-2)*(j)]*1.10136;
printf("%6.3f %12.6f\n",sourceprof,z);
fprintf(fp,"%6.3f %12.6f\n",sourceprof,z);

/* free_dmatrix(W,0,0);*/
free_dvector(b,0);

/* free_dmatrix(S,0,0);*/

```

```
free_dvector(profabs,0);
free_dmatrix(OHR,0,0);
free_dvector(dbC,0);
free_dvector(d1C,0);
free_dvector(dcC,0);
free_dvector(x,0);

}
free_dvector(OHR1,0);
free_dvector(R2,0);
free_dvector(PHI,0);
free_dvector(lay,0);
free_dvector(tool,0);
free_ivector(tooli,0);
free_dvector(Z1,0);

fclose(fp);
/* return EXIT_SUCCESS;*/

}
```

Programa de resistencias eléctricas esférico

```

/* Programa de Resistencias eléctricas esféricas
 * Ricardo Zavala Torres
 */

```

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <stddef.h>
#include <alloc.h>
#include "nrutil.h"
#define NR_END 0
#define FREE_ARG char*

```

```

#define P3 atan(1)*4

```

```

double *y;
double *Z;
double *R;
double *R2;
double *PHIT1;
double *PHIT2;
double **OHR;
double *OHRS;
double *B;
double *C;
double *D;
double *abd;
double *d1C;
double *dcC;
double *dbC;
double *b;
double *x;
double *tool;
double *lay;
int *jpot;

```

```

double RP(int i,int j)
{
double o,q,tesis;
o=(PHIT1[i]*P3/180/2);
q=(PHIT1[i+1]*P3/180/2);
tesis=((1/OHR[i][j])*log(tan((q+o)/4)/tan(o/2)))
/(2*P3*(R2[j+1]-R2[j]))
+ (((1/OHR[i+1][j])*log(tan(q/2)/tan((q+o)/4)))
/(2*P3*(R2[j+1]-R2[j])));

```

```

return tesis;
}
double RPZ(int i,int j)
{
double o,q,r,p,tesis;
o=(PHIT1[i]*P3/180/2);
p=tan(o);
q=(PHIT1[i+1]*P3/180/2);
r=tan(q);
tesis=((OHR[i][j]+OHR[i][j+1])/(2*(OHR[i][j]*OHR[i][j+1])))
*(log(r/p))
/(2*P3*(R2[j+1]-R2[j]));
return tesis;
}

```

```

double RSZ(int i,int j)
{
double z,w,y,v,rserie;

```

```

        z=(PHIT2[i]*P3/180);
        w=cos(z);
        y=(PHIT2[i+1]*P3/180);
        v=cos(y);
rserie=( (OHR[i][j]+OHR[j][j+1]) / (2*OHR[i][j]*OHR[j][j+1]))
          *( (R[j+1]-R[j])/(R[j]*R[j+1]))
          *(1/(2*P3*(w-v)) );

    return rserie;
}

double RS(int i,int j)
{
    double z,w,y,v,rserie;

        z=(PHIT2[i]*P3/180);
        w=cos(z);
        y=(PHIT2[i+1]*P3/180);
        v=cos(y);
rserie((((1/OHR[i][j]) * R[j+1])+(1/OHR[i][j+1]) *R[j] ))
        / (R[j]*R[j+1]))
        *( (R[j+1]-R[j]) / (R[j+1]+R[j]))
        *(1/(2*P3*( w-v )));

    return rserie;
}

void matrixZ(int I1,int J1,int ni)
{
    int i,j;
    FILE *fp;
    d1C=dvector(0,ni+1);
    dcC=dvector(0,ni+1);
    dbC=dvector(0,ni+1);
    b=dvector(0,ni+1);
    x=dvector(0,ni+1);

    for(i=1;i<I1-1;i++)
    {
        j=0;
        d1C[i+j+((I1-1)*j)+1]=1/RS(i,j)+1/RP(i-1,j)+1/RP(i,j);
        dbC[i+j+((I1-1)*j)+1]=-1/RP(i,j);
        dcC[i+j+((I1-1)*j)+1]=-1/RS(i,j);
    }

    for(j=1;j<J1-2;j++)
    {
        i=0;
        d1C[i+j+((I1-1)*j)+1]=1/RS(i,j)+1/RS(i,j-1)+1/RP(i,j);
        dbC[i+j+((I1-1)*j)+1]=-1/RP(i,j);
        dcC[i+j+((I1-1)*j)+1]=-1/RS(i,j);
    }
    j=J1-2;
    i=0;
        d1C[i+j+((I1-1)*j)+1]=1/RS(i,j)+1/RS(i,j-1)+1/RP(i,j);
        dbC[i+j+((I1-1)*j)+1]=-1/RP(i,j);
        dcC[i+j+((I1-1)*j)+1]=0.0;

    for(j=1;j<J1-2;j++)
    {
        i=I1-1;
        d1C[i+j+((I1-1)*j)+1]=1/RS(i,j)+1/RS(i,j-1)+1/RP(i-1,j);
        dbC[i+j+((I1-1)*j)+1]=0.0;
        dcC[i+j+((I1-1)*j)+1]=-1/RS(i,j);
    }
    j=J1-2;
    i=I1-1;

```

```

d1C[i+j+((I1-1)*j)+1]=1/RS(i,j)+1/RS(i,j-1)+1/RP(i-1,j);
dbC[i+j+((I1-1)*j)+1]=0.0;
dcC[i+j+((I1-1)*j)+1]=0.0;

for(i=1;i<J1-1;i++)
for(j=1;j<J1-2;j++)
{
d1C[i+j+((I1-1)*j)+1]=1/RS(i,j)+1/RS(i,j-1)+1/RP(i,j)+1/RP(i-1,j);
dbC[i+j+((I1-1)*j)+1]=1/RP(i,j);
dcC[i+j+((I1-1)*j)+1]=1/RS(i,j);
}
for(i=1;i<J1-1;i++)
{
j=J1-2;
d1C[i+j+((I1-1)*j)+1]=1/RS(i,j)+1/RS(i,j-1)+1/RP(i,j)+1/RP(i-1,j);
dbC[i+j+((I1-1)*j)+1]=1/RP(i,j);
dcC[i+j+((I1-1)*j)+1]=0.0;
}

i=0;
j=0;

d1C[i+j+((I1-1)*j)+1]=1/RS(i,j)+1/RP(i,j);
dbC[i+j+((I1-1)*j)+1]=1/RP(i,j);
dcC[i+j+((I1-1)*j)+1]=1/RS(i,j);

j=0;
i=I1-1;
d1C[i+j+((I1-1)*j)+1]=1/RS(i,j)+1/RP(i-1,j);
dbC[i+j+((I1-1)*j)+1]=0.0;
dcC[i+j+((I1-1)*j)+1]=1/RS(i,j);

for (j=1;j<=ni;j++)
{
b[j]=0.0;
x[j]=0.0;
if(j==(I1/2) {b[j]=1;
x[j]=1;
}

/*printf("%d,%10.5f,%10.5f,%10.5f,%10.5f\n",j,b[j],d1C[j],dbC[j],dcC[j]);
fprintf(fp,"%d,%10.5f,%10.5f,%10.5f,%10.5f\n",j,b[j],d1C[j],dbC[j],dcC[j]);*/

}

}

void malla(int I1,int J1,double F1,double F2,double PHI1,double RH1,double R1)
{
int i,j,k;
double D1,D2;

R=dvector(0,J1);
R2=dvector(0,J1);

PHIT1=dvector(0,I1);
PHIT2=dvector(0,I1);
jpot=ivector(0,5);
if (F1 == 1.0)
D1=PHI1/(I1-1);
else
D1=PHI1*(1-F1)/(1-pow(F1,I1));

if (F2 == 1.0)

```

```

        D2=RH1/(J1-1);
    else
        D2=RH1*(1-F2)/(1-pow(F2,J1));

    R[0]=R1;
    for(j=1;j<J1;j++)
    {
        R[j]=R[j-1]+D2;
        D2=F2*D2;
    }

    PHIT1[0]=1;
    for(i=1;i<I1;i++)
    {
        PHIT1[i]=PHIT1[i-1]+D1;
        D1=F1*D1;
    }
    PHIT1[I1-1]=PHI1;

    for (k=0;k<4;k++)
    {
        for(j=0;j<J1;j++)
        {
            if (R[j]<tool[k] && R[j+1]>tool[k])
            { R[j+1]=tool[k];
              jpot[k]=j+1;
            }
        }
        R[J1-1]=RH1;
    }

    PHIT2[0]=PHIT1[0];
    for (i=1;i<=I1;i++)
    {
        PHIT2[i]=(PHIT1[i]+PHIT1[i-1])/2;
        PHIT2[I1]=179;
    }
    R2[0]=R[0];
    for (j=1;j<J1;j++)
    {
        R2[j]=(R[j]+R[j-1])/2;
    }

    y=dvector(0,I1*J1);
    for(i=0;i<I1;i++)
    for(j=0;j<J1;j++)
    {
        y[i+I1*j]=R[j]*cos(PHIT1[i]*P3/180);
    }
}

void matriz(int ni,int I1,int J1)
{
    int i,j,k;
    FILE *fp;

    D=dvector(0,ni+1);
    d1C=dvector(0,ni+1);
    dcC=dvector(0,ni+1);
    dbC=dvector(0,ni+1);
    b=dvector(0,ni+1);
    x=dvector(0,ni+1);

```

```

for(i=0;i<ni;i++)
{
d1C[i+1]=0.0;
dbC[i+1]=0.0;
dcC[i+1]=0.0;
D[i]=0.0;
}

for (i=0;i<I1-1;i++)
{
D[i]=1/C[i];
}

for(k=0;k<J1-2;k++)
{
for (i=(k+1)*I1;i<((I1-1)*2)+1+(k*(I1));i++)
{
D[i]=1/C[i-(k+1)];
}
}

/*for(i=0;i<ni;i++)
{
printf("%d,%6.3f,%6.3f,%6.3f\n",i,D[i],B[i],C[i]);
fprintf(fp,"%d,%6.3f,%6.3f,%6.3f\n",i,D[i],B[i],C[i]);
}*/

for (i=0;i<I1;i++)
{
if (i==0) d1C[i+1]=1/B[i]+D[i];
else
if (i==(I1-1)) d1C[i+1]=1/B[i]+D[i-1];
else
d1C[i+1]=1/B[i]+D[i]+D[i-1];
}

for(k=0;k<J1-2;k++)
for (i=I1;i<ni;i++)
{
if(i==I1*(k+1)) d1C[i+1]=1/B[i-I1]+1/B[i]+D[i];
else
if(i==I1+(I1-1)*(k+1)+k) d1C[i+1]=1/B[i-I1]+1/B[i]+D[i-1];
else
d1C[i+1]=1/B[i]+1/B[i-I1]+D[i]+D[i-1];
}

for(i=0;i<ni;i++)
{
for(j=0;j<ni;j++)
{
if((i+j)==I1+(i*2))dcC[i+1]=-1/B[i];
if((i+j)==1+(i*2))dbC[i+1]=-D[i];
}
}

for(k=0;k<J1-1;k++)
for(i=0;i<ni;i++)
{

```

```

    if(i==(I1-1)+I1*k)
        dbC[i+1]=0.0;
}

for (j=1;j<=ni;j++)
{
    b[j]=0.0;
    x[j]=0.0;
    if(j==I1/2) {b[j]=1;
                 x[j]=1;
                }

    /* printf("%d,%10.5f,%10.5f,%10.5f,%10.5f\n",j,b[j],d1C[j],dbC[j],dcC[j]);
    fprintf(fp,"%d,%10.5f,%10.5f,%10.5f,%10.5f\n",j,b[j],d1C[j],dbC[j],dcC[j]);*/

}

}

void resistencias(int ni,int I1,int J1)
{
    int i,j;
    double z,w,y,v,o,q;
    B=dvector(0,ni+1);
    C=dvector(0,ni+1);
    for(i=0;i<ni;i++)
    {
        C[i]=0.0;
        B[i]=0.0;
    }
    /*for (i=0;i<I1;i++)
    {
        for(j=0;j<J1-1;j++)

        {
            B[(i+j)+(I1-1)*j]=RSZ(i,j);

        }
    }

    for (i=0;i<I1-1;i++)
    {
        for (j=0;j<J1-1;j++)
        {
            C[(i+j)+(I1-2)*j]=RPZ(i,j);
        }
    }

    */

    for(j=0;j<J1-1;j++)
    {
        for (i=0;i<I1;i++)B[(i+j)+(I1-1)*j]=RS(i,j);
        for (i=0;i<I1-1;i++)C[(i+j)+(I1-2)*j]=RP(i,j);
    }

}

void xyohr(int I1,int J1,int nlay,float sourceprof)
{
    int i,j,k;

```

```

Z=dvector(0,nlay);
OHR=dmatrix(0,J1,0,J1);
for(k=0;k<nlay-1;k++)
{
    Z[k]=sourceprof-lay[k];
}

for(k=0;k<nlay-1;k++)
{ if (Z[k]>=-tool[0] && Z[k]<=0)
    Z[k]=0.0;
for(i=0;i<J1;i++)
{
for(j=0;j<J1;j++)
{

if (y[i+1*j]>Z[0])
    OHR[i][j]=OHR[0];
if (Z[k]<0)
if (y[i+1*j]<=Z[k] && y[i+1*j]>Z[k+1])
    OHR[i][j]=OHR[k];
if (y[i+1*j]<=Z[k] && y[i+1*j]>Z[k+1])
    OHR[i][j]=OHR[k+1];
if (y[i+1*j]<Z[nlay-2])
    OHR[i][j]=OHR[nlay-1];

}
}
}
}

void nerror(char error_text[])
/* Numerical Recipes standard error handler */
{
    fprintf(stderr,"Numerical Recipes run-time error...\n");
    fprintf(stderr,"%s\n",error_text);
    fprintf(stderr,"...now exiting to system...\n");
    exit(1);
}

int *ivector(long nl, long nh)
/* allocate an int vector with subscript range v[nl..nh] */
{
    int *v;

    v=(int *)malloc((size_t)((nh-nl+1+NR_END)*sizeof(int)));
    if (!v) nerror("allocation failure in ivector()");
    return v-nl+NR_END;
}

double *dvector(long nl, long nh)
/* allocate a double vector with subscript range v[nl..nh] */
{
    double *v;

    v=(double *)malloc((size_t)((nh-nl+1+NR_END)*sizeof(double)));
    if (!v) nerror("allocation failure in dvector()");
    return v-nl+NR_END;
}

double **dmatrix(long nrl, long nrh, long ncl, long nch)
/* allocate a double matrix with subscript range m[nrl..nrh][ncl..nch] */
{
    long i, nrow=nrh-nrl+1, ncol=nch-ncl+1;
    double **m;

    /* allocate pointers to rows */
    m=(double **) malloc((size_t)((nrow+NR_END)*sizeof(double**)));
    if (!m) nerror("allocation failure 1 in matrix()");
    m += NR_END;
    m -= nrl;
}

```

```

/* allocate rows and set pointers to them */
m[nrl]=(double *) malloc((size_t)((nrow*ncol+NR_END)*sizeof(double)));
if (!m[nrl]) perror("allocation failure 2 in matrix()");
m[nrl] += NR_END;
m[nrl] -= ncl;

for(i=nrl+1;i<=nrh;i++) m[i]=m[i-1]+ncol;

/* return pointer to array of pointers to rows */
return m;
}

void free_ivector(int *v, long nl)
/* free an int vector allocated with ivector() */
{
    free((FREE_ARG) (v+nl-NR_END));
}

void free_dvector(double *v, long nl)
/* free a double vector allocated with dvector() */
{
    free((FREE_ARG) (v+nl-NR_END));
}

void free_dmatrix(double **m, long nrl, long ncl)
/* free a double matrix allocated by dmatrix() */
{
    free((FREE_ARG) (m[nrl]+ncl-NR_END));
    free((FREE_ARG) (m+nrl-NR_END));
}

int max0C(int num1,int num2)
{
    int regreso;
    if(num1>=num2)
        regreso=num1;
    else
        regreso=num2;

    return(regreso);
}

int min0C(int num1,int num2)
{
    int regreso;
    if(num1<=num2)
        regreso=num1;
    else
        regreso=num2;

    return(regreso);
}

double ddotC(int n, int la,int k,int N,int mu,int j)
{
    int i;
    double regreso;

    regreso=0;
    for(i=0;i<n;i++)
        regreso+=abd[(la+i)*N+k]*abd[(mu+i)*N+j];
    return(regreso);
}

double ddot2C(int n,int la, int k, int N,double *dy)
{
    int i;
    double regreso;

```

```

regreso=0;
for(i=0;i<n;i++)
{
    regreso+=abd[(la+i)*N+k]*(*dy);
    dy++;
}
return(regreso);
}

void daxpyC(int n,double da,int la,int k,int N, double *dy)
{
    int m,i,mp1;
    double *dyanterior;

    dyanterior=dy;

    m=n%4;
    for(i=1;i<=m;i++)
    {
        *dy=*dy+da*abd[(la+i-1)*N+k];
        dy++;
    }

    dy=dyanterior;
    if(n>=4)
    {
        mp1=m+1;
        dy+=m;
        for(i=mp1;i<=n;i++)
        {
            *dy=*dy+da*abd[(la+i-1)*N+k];
            dy++;
        }
    }
}

void adaptC(int jmax,int kmax)
{
    int i,k0,j,k,ik,jk,mu;
    int la,lm,lb,kb;
    int m,n,jmax2;
    double s,t;
    FILE *arch;

    /*write the b vector to be solved*/
    /* b=(double *)propio_malloc((kmax+1)*sizeof(double));

    for(i=1;i<=kmax;i++)
        b[i]=0;
    k0=(jmax+1)*(i0-1)+1;
    b[k0]=current;*/

    /*Band Storage*/
    m=jmax+1;
    jmax2=jmax+2;
    n=kmax;

    abd=dvector(0,(jmax2+1)*(n+1));

    /*zero the matrix*/
    for(j=1;j<=n;j++)
    {
        for(i=1;i<=jmax2;i++)
            abd[i*n+j]=0;
    }

    /*write in the non zero diagonals in band format*/
    for(i=1;i<=n;i++)
        abd[jmax2*n+i]=d1C[i];
}

```

```

for(i=2;i<=n;i++)
abd[m*n+i]=dbC[i-1];
for(i=jmax2;i<=n;i++)
abd[l*n+i]=dcC[i-m];

/*do the L D L factoring*/

for(j=1;j<=n;j++)
{
s=0;
ik=m+1;
jk=max0C(j-m,1);
mu=max0C(m+2-j,1);
if(m>=mu)
{
for(k=mu;k<=m;k++)
{
t=0;
t=abd[k*n+j]-ddotC(k-mu,ik,jk,n,mu,j);
t=t/abd[(m+1)*n+jk];
abd[k*n+j]=t;
s=s+(t*t);
ik--;
jk++;
}
}
/*aquí significa que m fue menor a mu */
s=abd[(m+1)*n+j]-s;
abd[(m+1)*n+j]=sqrt(s);
}/*j*/

/*solve trans(R)*y=b*/
for(k=1;k<=n;k++)
{
lm=min0C(k-1,m);
la=m+1-lm;
lb=k-lm;
t=ddot2C(lm,la,k,n,&b[lb]);
b[k]=(b[k]-t)/abd[(m+1)*n+k];
}

/*solve r*x=y*/

for(kb=1;kb<=n;kb++)
{
k=n+1-kb;
lm=min0C(k-1,m);
la=m+1-lm;
lb=k-lm;
b[k]=b[k]/abd[(m+1)*n+k];
t=b[k]*-1;
daxpyC(lm,t,la,k,n,&b[lb]);
}
free_dvector(abd,0);
}

main()
{

int I1,J1,i,j,n,zj,ni,l,k,m,t,ip,pi,N,aa,bb,kk,nlay,nmues;
float S9,L9,RH1,R1,PHI1,PHI3,F1,F2,D1,D2,A1,A2,A3,A4,OHR0,R6,R3,TANPHI,M1,C3,D3;
float z,w,v,sumaa,suma,sum,smax,r1,c,sumaaa,dphi,dr;
float o,q,analytical,error,xx;
float yy,zz;
float prof,sourceprof,start;
float sn,ln,lat1,lat2,lat,fro,res,end,samp;;
char name[14];
FILE *fp;
printf("Archivo a salvar:");

```

```

gets(name);
if (name[0]!='\0'){
    fprintf(stderr,"no filename typed\n");
    exit(EXIT_SUCCESS);
}
fp=fopen(name,"w");
if (fp==NULL){
    fprintf(stderr,"cannot open %s\n",name);
    exit(EXIT_FAILURE);
}

printf("number of nodes in r direction:\n");
scanf("%d",&J1);
printf("number of nodes in phi direction:\n");
scanf("%d",&I1);
fprintf(fp,"number of nodes in r direction: %d\n",J1);
fprintf(fp,"number of nodes in phi direction: %d\n",I1);

/* RH1 is the radius of dona and PHI1 is the angle of calculation zone */

/* internal radius of dona */

R1=0;

printf("internal radius of dona:\n");
scanf("%f",&R1);
printf("radial thickness of dona:\n");
scanf("%f",&RH1);
printf("angular zone of dona:\n");
scanf("%f",&PHI1);
fprintf(fp,"internal radius of dona: %f\n",R1);
fprintf(fp,"radial thickness of dona: %f\n",RH1);
fprintf(fp,"angular zone of dona: %f\n",PHI1);

/* *P3/180; */

/* F1 and F2 expansion factors in r and phi direction */

printf("Enter Grid-expansion factor in radial:\n");
scanf("%f",&F2);
printf("Enter Grid-expansion factor in Phi:\n");
scanf("%f",&F1);
fprintf(fp,"Enter Grid-expansion factor in radial: %f\n",F2);
fprintf(fp,"Enter Grid-expansion factor in Phi: %f\n",F1);

tool=dvector(0,5);
ni=(J1-1)*I1;
/*
I1=12;
J1=20;
R1=0;
R1=0.01;
RH1=50;
PHI1=180;
F2=1.4;
F1=1;
*/
printf("Enter number of layers:\n");
scanf("%d",&nlay);
fprintf(fp,"Enter number of layers: %d\n",nlay);

/*nlay=3;*/

```

```

lay=dvector(0,nlay);
OHRS=dvector(0,nlay);

tool[0]=0.4064;
tool[1]=1.6256;
tool[2]=5.2832;
tool[3]=6.0960;
tool[4]=5.58;

/*
lay[0]=1002.54;
lay[1]=1005.08;
lay[2]=1007.62;
lay[3]=1010.16;

OHRS[0]=0.0714;
OHRS[1]=0.01;
OHRS[2]=0.2;
OHRS[3]=0.05;
OHRS[4]=0.03;
*/
/*lay[0]=1010;
lay[1]=1020;

OHRS[0]=0.5;
OHRS[1]=0.02;
OHRS[2]=0.5;

start=1000; */

printf("start:\n");
scanf("%f",&start);
fprintf(fp,"start: %f\n",start);
printf("end:\n");
scanf("%f",&end);
fprintf(fp,"end: %f\n",end);
printf("samp rate:\n");
scanf("%f",&samp);
fprintf(fp,"samp rate: %f\n",samp);

/*sourceprof=1030;*/
/*printf("prof=");
scanf("%f",&sourceprof);*/
for(i=0;i<nlay;i++)
{
    printf("Enter resistivity of layer %d:\n",i+1);
    scanf("%f",&res);
    fprintf(fp,"resistivity of layer %d: %f\n",i+1,res);
    OHRS[i]=1/res;
}
for(k=0;k<nlay-1;k++)
{
    printf("Enter depth of boundary %d:\n",k+1);
    scanf("%f",&fro);
    fprintf(fp,"depth of boundary %d: %f\n",k+1,fro);
    lay[k]=fro;
}
malla(11,J1,F1,F2,PHI1,RH1,R1);
nmues=(end-start)/samp;
for(kk=0;kk<=nmues;kk++)
{
    sourceprof=start+(kk*samp);
    xyohr(11,J1,nlay,sourceprof);

/*
for (i=0;i<11;i++)
{
    for (j=0;j<11;j++)
    {

```

```

        printf("%4.2f",OHR[i][j]);
        fprintf(fp,"%4.2f",OHR[i][j]);
    }
    printf("\n");
    fprintf(fp,"\n");
}

*/
resistencias(ni,l1,l1);
matriz(ni,l1,l1);
/*matrixZ(l1,l1,ni);*/
adaptC(l1-1,ni);

/*
printf("solucion: Radio y potencial\n");
fprintf(fp,"solucion: Radio y Potencial\n");
for (i=1;i<=N;i++){ printf("%d %6.5f\n",i,b[i]);

                    fprintf(fp,"%d %6.5f\n",i,b[i]);}*/

/*fprintf(fp,"\n");
printf("\n");*/

for (i=0;i<ni;i++){
    b[i]=b[i+1];
}

/*
fprintf(fp,"i, j, z, r, x, d1, d2, d3, b\n");
for (i=0;i<l1;i++)
{
    for (j=0;j<J1-1;j++)
    {
        xx=R[j]*sin(PHIT1[i]*P3/180);
        yy=R[j]*cos(PHIT1[i]*P3/180);
        z=b[i+j+((l1-1)*j)];

        zz=d1C[i+j+((l1-1)*j)+1];
        analytical=dbC[i+j+((l1-1)*j)+1];
        error=dcC[i+j+((l1-1)*j)+1];

        printf("%d %d %6.4f %8.6f %8.6f %6.4f %8.6f %8.6f %6.4f \n",i,j,xx,yy,z,zz,analytical,error,x[i+j+((l1-1)*j)+1]);
        fprintf(fp,"%d %d %6.4f %8.6f %8.6f %6.4f %8.6f %8.6f %6.4f \n",i,j,xx,yy,z,zz,analytical,error,x[i+j+((l1-1)*j)+1]);
    }
}

printf("No x y pot rhoapp rad anal error\n");
fprintf(fp,"No x y pot rhoapp anal error rad\n");
for (j=0;j<J1;j++)

{ i=l1-1;
for (i=0;i<l1;i++)
{
    if (j==J1-1) b[i+j+((l1-1)*j)]=0.0;
    xx=R[j]*sin(PHIT1[i]*P3/180);
    yy=R[j]*cos(PHIT1[i]*P3/180);
    z=b[i+j+((l1-1)*j)];
    zz=R[j]*4*P3*b[i+j+((l1-1)*j)];
    analytical=(1/0.02)/(4*P3*R[j]);
    error=100-(z/analytical)*100;

    printf("%d %6.4f %8.6f %8.6f %6.4f %8.6f %8.6f %6.4f \n",(i+j)+(l1-1)*j,xx,yy,z,zz,analytical,error,R[j]);
    fprintf(fp,"%d %6.4f %8.6f %8.6f %6.4f %8.6f %8.6f %6.4f \n",(i+j)+(l1-1)*j,xx,yy,z,zz,analytical,error,R[j]);
}
}
*/

```

```

/* i=8;
j=2;*/
/*xx=R[j]*sin(PHIT1[j]*P3/180);
y=R[j]*cos(PHIT1[j]*P3/180);
z=x[i+j+((I1-1)*j)];
zz=R[j]*4*P3*x[i+j+((I1-1)*j)];*/

/* zz=R[j]*4*P3*b[i+j+((I1-1)*j)];*/

/* analytical=(1/OHR0)/(4*P3*R[j]);
error=100-(z/analytical)*100;*/

/*printf("%d %6.4f %8.6f %8.6f %6.4f %8.6f %8.6f %6.4f \n",(i+j)+((I1-1)*j),xx,y,z,analytical,error,R[j] );
fprintf(fp,"%d %6.4f %8.6f %8.6f %6.4f %8.6f %8.6f %6.4f \n",(i+j)+((I1-1)*j),xx,y,z,analytical,error,R[j] );*/
/* printf("%6.4f %8.6f \n",sourceprof,zz);
fprintf(fp,"%6.4f %8.6f \n",sourceprof,zz);*/

i=I1-1;
j=jpot[0];
/*j=3;*/
sn=R[j]*4*P3*b[i+j+((I1-1)*j)];
j=jpot[1];
/*j=5;*/
ln=R[j]*4*P3*b[i+j+((I1-1)*j)];
j=jpot[3];
/*j=8;*/
lat1=b[i+j+((I1-1)*j)];
j=jpot[2];
/*j=7;*/
lat2=b[i+j+((I1-1)*j)];
lat=497.92987*(lat2-lat1);
printf("%6.4f %8.6f %8.6f %8.6f \n",sourceprof,sn,ln,lat);
fprintf(fp,"%6.4f %8.6f %8.6f %8.6f \n",sourceprof,sn,ln,lat);

free_dvector(dcC,0);
free_dvector(dbC,0);
free_dvector(d1C,0);
free_dvector(b,0);
free_dvector(x,0);
free_dvector(B,0);
free_dvector(C,0);
free_dvector(D,0);
free_dvector(Z,0);

free_dmatrix(OHR,0,0);

}

free_dvector(lay,0);
free_dvector(OHRS,0);
free_dvector(tool,0);
free_ivector(jpot,0);
free_dvector(R,0);
free_dvector(R2,0);
free_dvector(PHIT1,0);
free_dvector(PHIT2,0);
free_dvector(y,0);

fclose(fp);
}

```

Programa solución analítica 3 capas

```

/* program theoretical equation of apparent resistivity
 * Ricardo Zavala Torres
 */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <stddef.h>
#define nsize 250
#define n 10
#define P3 atan(1)*4

float r11(float Z,float h,float k12,float k23)
{
    int k,i;
    float a12m,k21,a,sum,sum1a,appar;
    k21=-k12;
    a12m=fabs(Z)-1;

    sum=0;
        for(k=1;k<=10;k++)
        {
            a=pow(k21*k23,k-1)/fabs(k*h+a12m);
            sum=sum+a;
        }

        appar=(1+k12/fabs(a12m)+((1-k12)*k23*sum));
        return appar;
}

float r22(float Z,float h,float k12,float k23)
{
    int k,i;
    float a12m,k21,a,b,c,d,sum,sum1a,appar,suma,sumb,sumc;
    k21=-k12;
    a12m=fabs(Z)+1;

    sum=0;
    suma=0;
    sumb=0;
    sumc=0;

        for(k=1;k<=10;k++)
        {
            a=pow(k21*k23,k-1)/fabs((k-1)*h+a12m);
            sum=sum+a;
        }

        for(k=1;k<=10;k++)
        {
            b=pow(k21*k23,k-1)/fabs(k*h-a12m);
            suma=suma+b;
        }

        for(k=1;k<=10;k++)
        {
            c=pow(k21*k23,k)/fabs(k*h-1);
            sumb=sumb+c;
        }

        for(k=1;k<=10;k++)
        {
            d=pow(k21*k23,k)/fabs(k*h+1);
            sumc=sumc+d;
        }

        appar=(1+k21*sum+k23*suma+sumb+sumc);
        return appar;
}

```

```

float r21(float Z,float h,float k12,float k23)
{
    int k,i;
    float a12m,k21,a,b,sum,sum1a,appar,suma;
    k21=-k12;
    a12m=fabs(Z)-1;

    sum=0;
    suma=0;

        for(k=1;k<=10;k++)
        {
            a=pow(k21*k23,k-1)/fabs(k*h+a12m);
            sum=sum+a;
        }
        for(k=1;k<=10;k++)
        {
            b=pow(k21*k23,k)/fabs(k*h+1);
            suma=suma+b;
        }

        appar=(1-k12)*(1+k23*sum+suma);
        return appar;
}
float r31(float Z,float h,float k12,float k23)
{
    int k,i;
    float a12m,k21,a,sum,sum1a,appar,suma;
    k21=-k12;

    sum=0;

        for(k=1;k<=10;k++)
        {
            a=pow(k21*k23,k)/fabs(k*h+1);
            sum=sum+a;
        }

        appar=(1-k12)*(1-k23)*(1+sum);
        return appar;
}
float *vector(int ni)
{
    float *m;
    m=(float *) malloc(ni*sizeof(float));
    if (!m)
        { printf("Error de asignaci3n de memoria en matriz\n");
          abort();
        }
    return m;
}
main()
{
    float *appa;

    float *vector(int);
    float *bound;

    float *sourceprof;
    float k12,k23,k21,OHR1,OHR2,OHR3,L,h1,h,a,sum,sum1,sum2,b,summ,sum3,summm,c
    ,summmm,sum4,d,start,sum1a,Z,Z1,d0;
    int k,i;
    char name[14]={"poten2lc.dat"};
    FILE *fp;
    fp=fopen(name,"w");
    OHR1=1;
    OHR2=500;
    OHR3=2;
    k12=(OHR2-OHR1)/(OHR2+OHR1);
    k23=(OHR3-OHR2)/(OHR2+OHR3);
}

```

```

L=0.4064;
d0=0.2032;
h1=10;
appa=vector(nsize);

sourceprof=vector(nsize);
bound[0]=1010;
bound[1]=1020;
start=1000;

for (i=0;i<=240;i++)
{
    sourceprof[i]=start+(i*0.125);
    Z1=sourceprof[i]-bound[0];

    Z=-Z1;
    h=h1;
    if (Z1>=0)
    {
        appa[i]=OHR1*L*(((8*(-Z)+P3*d0)*(8*(h+Z)+P3*d0))/
            ((4*h+P3*d0)*pow(d0,2)));
    }
    else
    if (Z1<=-L)
    {
        appa[i]=OHR1*(1+L/(2*Z-L))*((
            (2*L*d0)/(Z*(4*h+P3*d0))
            )*
            atan( d0/ (2*(Z-L))
            ));
    }

    if (Z1>=-L && Z1<0)
    {
        appa[i]=OHR1*L*(8*(h+Z-L)+P3*d0)/(Z*(4*h+P3*d0));
    }

    if (Z1>=h1-L && Z1<h1)
    {
        appa[i]=OHR1*L*(8*(h+(-Z-h1+L)-L)+P3*d0)/((-Z-h1+L)*(4*h+P3*d0));
    }

    if (Z1>=h1)
    {
        appa[i]=OHR1*(1+L/(2*(-Z-h1+L)-L))*((
            (2*L*d0)/((-Z-h1+L)*(4*h+P3*d0))
            )*
            atan( d0/ (2*(-Z-h1+L))
            ));
    }
    fprintf(fp, "%6.3f,%6.3f,%6.3f\n", sourceprof[i], appa[i], Z1);
    printf("%6.3f,%6.3f,%6.3f\n", sourceprof[i], appa[i], Z1);
}
fclose(fp);
}

```

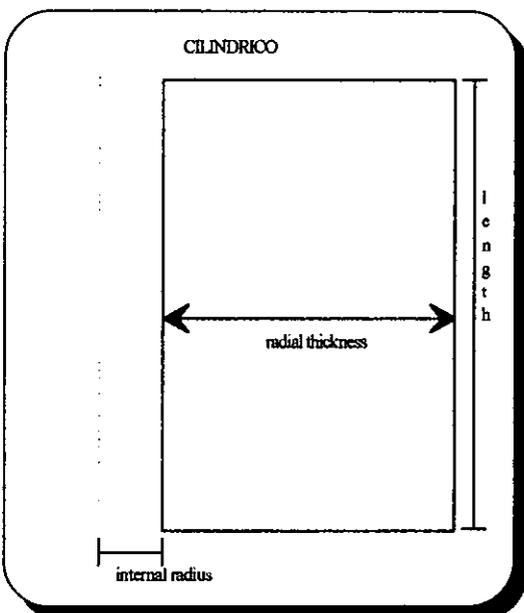
**Utilización de los programas de modelado de geometría
cilíndrica y esférica**

Utilización de la malla cilíndrica

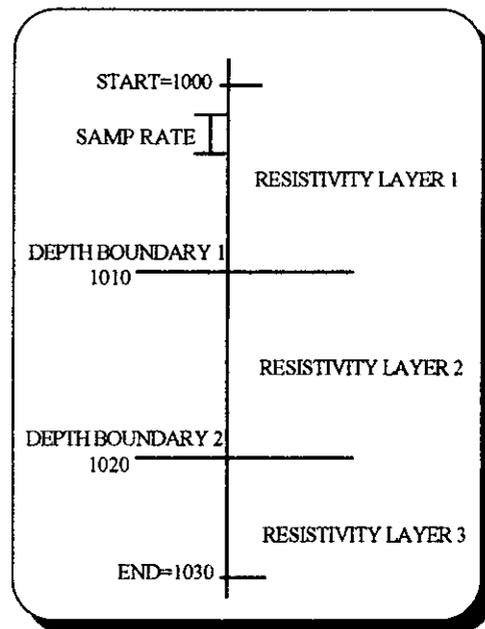
number of nodes in r direction: 12
number of nodes in z direction: 31
internal radius of cylinder: 0.010000
radial thickness of cylinder: 50.000000
length of cylinder: 100.000000
Enter Grid-expansion factor in radial: 1.800000
Enter Grid-expansion factor in z: 1.200000
Enter number of layers: 3
start: 1000.000000
end: 1030.000000
samp rate: 0.125000
resistivity of layer 1: 2.000000
resistivity of layer 2: 50.000000
resistivity of layer 3: 2.000000
depth of boundary 1: 1010.000000
depth of boundary 2: 1020.000000

Salida:

depth	apparho
1000.000	2.034003
1000.125	2.034003
1000.250	2.034003
1000.375	2.034003
1000.500	2.034003
1000.625	2.034003
1000.750	2.034003



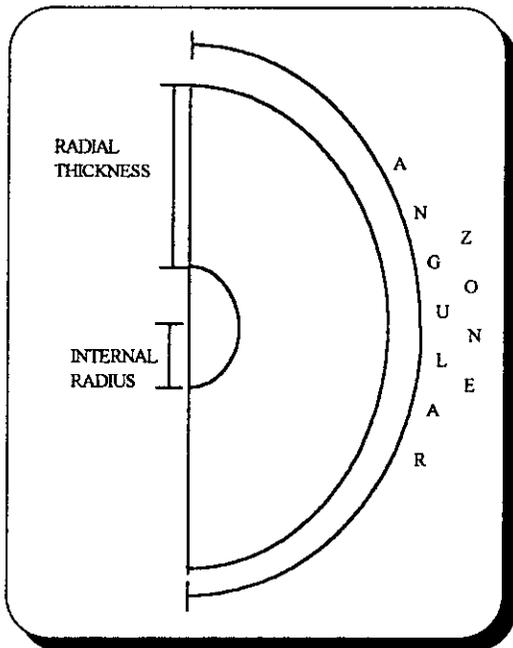
Parámetros de la malla cilíndrica



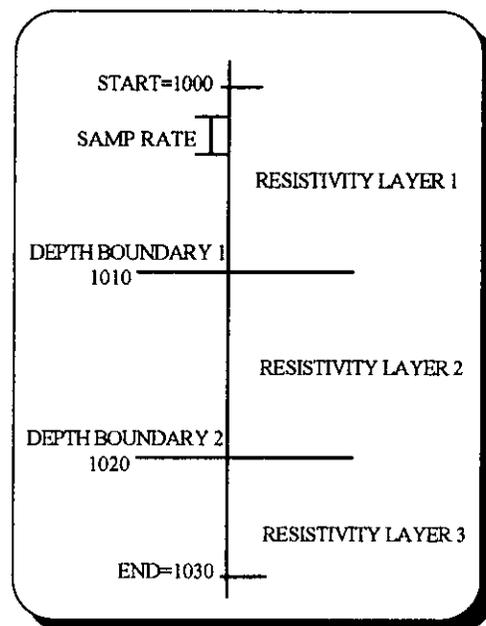
Datos del registro

Utilización de la malla esférica

number of nodes in r direction: 20
number of nodes in phi direction: 14
internal radius of dona: 0.010000
radial thickness of dona: 50.000000
angular zone of dona: 180.000000
Enter Grid-expansion factor in radial: 1.260000
Enter Grid-expansion factor in Phi: 1.000000
Enter number of layers: 3
start: 1000.000000
end: 1030.000000
samp rate: 0.125000
resistivity of layer 1: 2.000000
resistivity of layer 2: 50.000000
resistivity of layer 3: 2.000000
depth of boundary 1: 1010.000000
depth of boundary 2: 1020.000000
Salida:
depth apparho
1000.0000 1.876753
1000.1250 1.876753
1000.2500 1.876753
1000.3750 1.876753
1000.5000 1.876753
1000.6250 1.876753



Parámetros de la malla esférica



Datos del registro