Manual del usuario

SCEPTRE — SC1/SD2

Un programa para el análisis y la simulación de redes eléctricas no lineales y sistemas dinámicos

Wolf-Rainer Novender

Original: revisión del 4 de abril de 2026

Traducción parcial al español

Basada en el manual alemán; los capítulos se incorporan tras su revisión.

© 2026 Wolf-Rainer Novender, obra original
https://www.thm.de/iem/novender/sceptre

Índice general

1. Prólogo **1**
1.1. Panorama de prestaciones . . . 1
1.2. Historia del desarrollo . . . 2
1.3. Ejemplo . . . 2
1.4. Notas sobre este manual . . . 3

2. Instrucciones de SCEPTRE **5**
2.1. Convenciones del lenguaje . . . 5
2.2. Preparación . . . 6
2.3. Formato de entrada . . . 8
2.4. Estructura de la entrada . . . 8
2.5. `ELEMENTS` . . . 9
2.5.1. Forma general de los elementos de dos terminales . . . 9
2.5.2. Inductancias mutuas . . . 10
2.5.3. Fuentes dependientes de resistencias . . . 11
2.5.4. Fuentes de corriente controladas . . . 11
2.5.5. Derivadas temporales de las fuentes . . . 11
2.5.6. Llamada a un modelo . . . 11
2.5.7. Elementos con límites de tolerancia . . . 11
2.5.8. Fuentes en el análisis de corriente alterna . . . 12
2.6. `DEFINED PARAMETERS` . . . 12
2.7. `OUTPUTS` . . . 13
2.8. `INITIAL CONDITIONS` . . . 14
2.9. `FUNCTIONS` . . . 15
2.9.1. Instrucción `EQUATION` . . . 15
2.9.2. Instrucción `TABLE` . . . 15
2.10. `RUN CONTROLS` . . . 16
2.10.1. Análisis de corriente continua (DC) . . . 16
2.10.2. Análisis de corriente alterna (AC) . . . 18
2.10.3. Análisis transitorio (TR) . . . 18
2.10.4. Análisis DC seguido de TR . . . 19
2.10.5. Análisis DC seguido de AC . . . 19
2.10.6. Análisis AC seguido de TR . . . 19
2.10.7. Control de la salida . . . 19
2.10.8. Ejecuciones repetidas . . . 19
2.10.9. Control de la ejecución . . . 20
2.10.10. Ayudas para las pruebas . . . 20
2.11. `SENSITIVITY`, `MONTE CARLO`, `WORST CASE` y `OPTIMIZATION` . . . 20
2.12. Modelos . . . 21
2.12.1. Descripción de un modelo . . . 22
2.12.2. Inserción de un modelo . . . 23
2.13. `RERUN DESCRIPTION` . . . 24
2.14. `CONTINUE` . . . 27
2.15. `RE-OUTPUT` . . . 28

2.16. END 28

3. Indicaciones generales **29**
3.1. Diagnóstico de errores 29
3.2. Algunas reglas básicas 29
3.3. Inductancias y capacidades dependientes del tiempo 30
3.4. Tablas 30
3.5. Instrucciones aritméticas 30
3.6. Restricciones topológicas 30
3.6.1. Análisis DC, AC y TR 30
3.6.2. Análisis DC 30
3.7. «Computational delay» 31
3.8. Elección del método de integración 31
3.9. Definición de fuentes dependientes 32
3.10. Medición de corrientes y tensiones 33
3.11. Uso de subprogramas FORTRAN propios 33
3.12. Notación vectorial 34
3.13. Elección de las variables de estado 34
3.14. Cálculo de integrales 34
3.15. Estadísticas de la simulación 35
3.16. Salida impresa 35
3.17. Condiciones iniciales 35
3.17.1. Punto de trabajo 35
3.17.2. Integración implícita 36
3.17.3. Repeticiones 37
3.18. Funciones periódicas 37
3.18.1. Funciones ideales 38
3.18.2. FPULSE 38
3.18.3. FSINSQ 39
3.19. Nombres de variables reservados 39
3.20. Fuentes de corriente alterna 40

4. Ejemplos **41**
Los ejemplos como parte viva del libro 41
4.1. A01: inversor con carga RC 42
4.2. A02: seguidor de emisor con transformador 45
4.3. A03: circuito Darlington (análisis DC) 48
4.4. A04: circuito equivalente de pequeña señal 50
4.5. A05: sistema de ecuaciones diferenciales 52
4.6. Análisis de tolerancias de un circuito Darlington 54
4.6.1. A07: análisis Monte Carlo 54
4.6.2. A08: análisis de sensibilidad 56
4.6.3. A09: análisis de peor caso 59
4.6.4. A10: optimización 61
4.6.5. A11: circuito equivalente de pequeña señal (AC) 63
4.6.6. A13: actuador electrohidráulico 66

I Descripción matemática (Volumen II) 69

5. La teoría de SCEPTRE 70
5.1. Panorama general . . . 70
5.2. Planteamiento general de la solución . . . 70
5.3. Solución transitoria . . . 72
5.3.1. Operaciones matriciales . . . 72
5.3.2. Esquema de solución . . . 74
5.3.3. Derivadas temporales de las fuentes . . . 75
5.3.4. Fuentes linealmente dependientes . . . 75
5.3.5. Métodos de integración . . . 77
5.4. Análisis de corriente continua (DC) . . . 84
5.4.1. Método de Newton–Raphson . . . 84
5.4.2. Monte Carlo . . . 88
5.4.3. Sensibilidad . . . 88
5.4.4. Análisis del peor caso . . . 89
5.4.5. Optimización . . . 90
5.5. Análisis de corriente alterna (AC) . . . 91
5.5.1. Caso 1: solo fuentes independientes . . . 92
5.5.2. Caso 2: circuitos con fuentes linealmente dependientes . . . 94
5.5.3. Caso 3: circuitos que requieren derivadas temporales de fuentes independientes . . . 95
5.5.4. Caso 4: fuentes linealmente dependientes y derivadas de fuentes independientes . . . 96

A. Flujo del programa 97

B. Circuitos equivalentes de diodos y transistores 105
B.1. Diodo . . . 105
B.2. Transistor pnp . . . 106
B.3. Transistor npn . . . 107

C. Apéndices del volumen I [2] 108
C.1. Resumen de las instrucciones de SCEPTRE . . . 108
C.2. Variables internas: bloque `COMMON /CNTRLS/` . . . 109
C.3. Capacidad del programa . . . 110

D. Apéndices del volumen II [3] 111
D.1. Matrices dispersas . . . 111
D.2. Rutinas ensamblador para entrada y salida . . . 111
D.3. Método de integración implícita . . . 112
D.4. Deducción de la matriz B . . . 112
D.5. Evaluación de la característica del diodo al calcular condiciones iniciales . . . 112
D.6. Método de Newton–Raphson . . . 112
D.7. Identidad de las ecuaciones (5.79) y (5.80) . . . 112
D.8. Red adjunta . . . 112

E. Más explicaciones de SCEPTRE de Novender **113**
E.1. Elementos de convolución . . . 113
E.2. Núcleos de convolución . . . 114
E.3. Búferes de convolución . . . 114
E.4. Compresión del historial . . . 115
E.5. Derivadas parciales . . . 115
E.6. Notación vectorial . . . 116
E.7. Monitor del terminal . . . 116
E.8. Listados FULIST . . . 117
E.9. Listados WRITE DEBUG . . . 117
E.10. Derivadas de fuentes . . . 118
E.11. Formatos binarios heredados . . . 119
E.12. Funciones Fortran auxiliares . . . 119
E.13. Condiciones iniciales incompatibles . . . 120
E.14. Capacidades e inductancias variables . . . 121
E.15. Modelos permanentes . . . 122
E.16. D01: filtro y repeticiones AC . . . 123

F. Maestría en Energías Renovables de la UNSa **125**
F.1. Marco institucional y resoluciones . . . 125
F.2. Estructura del Plan 2021 . . . 125
F.3. Transformaciones Energéticas: contenidos mínimos . . . 126
F.4. Programa analítico de Transformaciones Energéticas . . . 126
F.5. Calendario y relación con la simulación . . . 126

Bibliografía **126**

1 Prólogo

El programa SCEPTRE se desarrolló para analizar redes eléctricas, con especial atención a los efectos no lineales. Su nombre significa *System for Circuit Evaluation and Prediction of Transient Radiation Effects* [1, 2, 3]. Este manual describe el uso y la teoría de la versión original de SCEPTRE. La versión revisada del programa lleva el nombre provisional SC1-SD2 y se describe en [4].

1.1. Panorama de prestaciones

SCEPTRE ofrece las siguientes funciones:

- análisis de corriente continua (DC), incluidos análisis de peor caso, sensibilidad y Monte Carlo, y optimización;
- análisis transitorio (TR);
- análisis de corriente alterna (AC);
- combinación de tipos de análisis (DC+TR y DC+AC);
- método de variables de estado;
- generación automática de las ecuaciones de red a partir de la topología y los elementos del circuito;
- cálculo de redes no lineales: cada elemento no lineal puede describirse mediante un modelo de complejidad arbitraria; los elementos pueden expresarse como constantes, tablas, expresiones algebraicas o funciones de otras magnitudes de la red;
- entrada de datos de la red y de instrucciones de control en formato libre, con instrucciones fáciles de recordar y sin necesidad de conocimientos de programación;
- biblioteca de modelos: los circuitos o subcircuitos de uso frecuente pueden guardarse de manera permanente e incorporarse a la red bajo estudio mediante una instrucción;
- determinación del punto de operación: el análisis DC puede combinarse con el análisis AC o TR para determinar el punto de operación de un circuito;
- ejecuciones repetidas de análisis con parámetros modificados;
- parámetros definidos por el usuario, que no tienen por qué estar relacionados con la red; también se puede especificar su derivada temporal, lo que permite integrar directamente sistemas de ecuaciones diferenciales sin describir una red;
- salida tabular o mediante gráficos impresos; varias magnitudes pueden representarse en un mismo gráfico, incluso cuando la abscisa no es el tiempo; existe una interfaz para programas gráficos externos;

- fuentes linealmente dependientes: se admiten tensiones y corrientes que dependen linealmente de corrientes en resistencias e inductancias y/o de tensiones en resistencias y condensadores[1], sin introducir un error por retardo computacional (*computational delay*);
- interrupción automática de la simulación en función del comportamiento de la red.

La posibilidad de definir parámetros y sus derivadas temporales es una de las características más destacadas de SCEPTRE. Permite resolver directamente sistemas de ecuaciones diferenciales y, con ello, simular problemas no eléctricos e interdisciplinarios. También se pueden simular conjuntamente varios circuitos que no estén conectados galvánicamente entre sí, es decir, que no compartan un potencial de referencia. Así se puede comparar el comportamiento de dos o más circuitos en una sola ejecución. SCEPTRE dispone además de una interfaz para subprogramas FORTRAN propios. Estas y otras características amplían su generalidad frente a otros programas de simulación [5].

1.2. Historia del desarrollo

IBM desarrolló SCEPTRE en 1966 por encargo del gobierno de Estados Unidos. Originalmente se concibió para la familia de computadoras IBM 7090/94. Años después, AFWL (Air Force Weapons Laboratory, AFB, Nuevo México, Estados Unidos) puso el programa a disposición gratuitamente para los sistemas IBM S/360 y CDC 6000/7000. Para preservar su portabilidad, SCEPTRE se programó íntegramente en FORTRAN IV. En la versión S/360, algunas operaciones de entrada y salida y rutinas de manipulación de bits y bytes se escribieron en ensamblador (véase el apartado D.2). IBM ofreció una evolución de SCEPTRE como programa bajo licencia con el nombre ASTAP [6].

La versión revisada SC1 es casi idéntica a la original y se describe con detalle en [4]. Fue programada íntegramente en Fortran 95[2], junto con la interfaz gráfica SD2. La sintaxis de SCEPTRE se simplificó en lo referente al uso de mayúsculas y minúsculas, pero sigue siendo compatible con la escritura original en mayúsculas.

1.3. Ejemplo

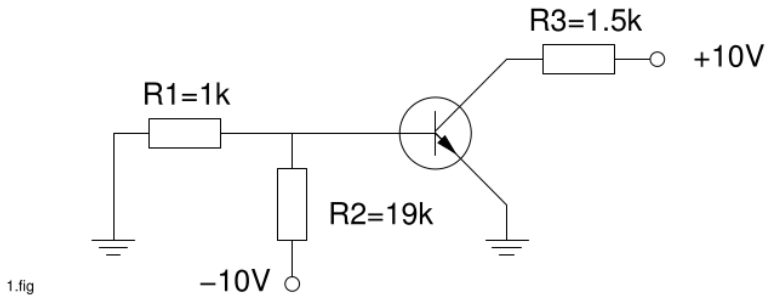


Figura 1.1: Circuito inversor sencillo. Figura reproducida del manual original.

El ejemplo muestra cómo simular el circuito de la figura 1.1. Se supone que el circuito equivalente del transistor 2N914A ya está guardado en la biblioteca de modelos. La figura

[1] Estas últimas se consideran variables de estado.
[2] https://gcc.gnu.org/fortran/

1.2 muestra el circuito preparado para SCEPTRE. Después de elegir un sistema de unidades adecuado, la red se describe mediante las siguientes instrucciones:

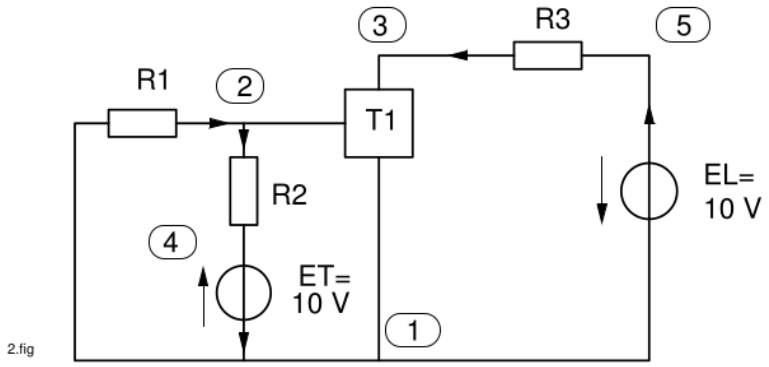


Figura 1.2: Representación del circuito inversor en SCEPTRE. Figura reproducida del manual original.

```
CIRCUIT DESCRIPTION
ELEMENTS
R1, 1-2 = 1
R2, 2-4 = 19
R3, 5-3 = 1.5
EL, 1-5 = 10
ET, 4-1 = 10
T1, 2-1-3 = MODEL 2N914A (PERM)
OUTPUTS
VR3, VR1, VCXT1, PLOT
RUN CONTROLS
STOP TIME = 500, RUN INITIAL CONDITIONS
END
```

Primero se determina el punto de operación del circuito (`RUN INITIAL CONDITIONS`). Con esos valores se simula su comportamiento transitorio durante 500 ns (`STOP TIME = 500`). Se obtienen, tanto en forma tabular como en un gráfico impreso, las tensiones en las resistencias R_1 y R_3 (`VR3`, `VR1`) y en el condensador C_X perteneciente al circuito equivalente del transistor T_1 (`VCXT1`).

Obsérvese que los valores de las magnitudes físicas (tensión, corriente, resistencia, capacitancia, tiempo, etc.) se asignan sin indicar unidades. Tampoco hay unidades predeterminadas: el usuario debe elegir un sistema de unidades cómodo pero coherente (véanse el apartado 2.2 y la tabla 2.3 del original).

1.4. Notas sobre este manual

Este manual se basa en la documentación original en inglés [2, 3], pero no pretende ser una traducción completa. En algunos lugares se añadieron datos procedentes de la experiencia del autor; en otros se omitieron pasajes o se remitió al lector a la documentación original. Por razones históricas, se conservaron en parte indicaciones específicas de las computadoras de la época. Después de todo, han pasado más de cincuenta años desde el

desarrollo de SCEPTRE, durante los cuales el hardware informático avanzó de manera extraordinaria.

Siguiendo la convención del inglés, este manual escribe los números decimales con punto y no con coma.

2 Instrucciones de SCEPTRE

2.1. Convenciones del lenguaje

En este manual, la forma general de una instrucción SCEPTRE se presenta enmarcada. Todas las palabras clave deben escribirse en mayúsculas; por ejemplo:

```
RUN INITIAL CONDITIONS ONLY
```

Los parámetros que debe proporcionar el usuario se escriben en cursiva; por ejemplo, *M name, L name1 – L name2*. Las llaves indican alternativas, como `DE` o `DJ` en la forma siguiente:

```
{DE | DJ} name = value
```

Los corchetes indican datos opcionales:

```
variable1 [, variable2...], PLOT (variable3)
```

Los valores predeterminados aparecen subrayados o se indican junto al recuadro. En diversos lugares se usa el término «tarjeta»; equivale a una línea de un archivo.

Para simplificar la exposición se adoptan las siguientes convenciones. En general se conservan los términos ingleses del lenguaje:

`element name`: Nombre de un componente de la red. Ejemplos: `R102`, `CLE`, `E1`, `LCOIL`.

`node`: Nombre de un nodo. Ejemplos: `0`, `GND`, `MASSE`, `EGON`.

`number`: Constante numérica en formato D, E, I o F, con o sin signo, de hasta 13 caracteres. Ejemplos: `314`, `3.14`, `-.314`, `3.14E-3`.

`constant`: Igual que `number`, pero siempre con punto decimal.

`value`: Puede ser un `number`, un parámetro definido, una `TABLE`, una `EQUATION`, una `EXPRESSION` o una función externa.

`special value`: Puede ser un `value`; el producto de una `constant` por una corriente `IR...` o una tensión `VR...` (por ejemplo, `3.*IRL` o `.5E-7*VR2`); el producto de un `value` por `J...` (por ejemplo, `PS3*JEIN` o `T43(VLS)*JAUS`); una `DIODE TABLE`; o una `DIODE EQUATION`.

`variable`: Puede ser una tensión o corriente (`V...`, `I...`; por ejemplo, `VRL`, `IRL`, `VJEIN`, `IEIN`); una fuente o su derivada (`E...`, `DE...`, `J...`, `DJ...`); un parámetro definido; el valor de un componente (`R1`, `CES`, `ML1L2`); `TIME`; o un parámetro interno de SCEPTRE (`XSTPSZ`, `XIR`, `XTMON`).

`TABLE name(variable)`. Una magnitud de la red también puede darse en forma tabular. La tabla se define bajo `FUNCTIONS`. `variable` es la variable independiente; si se omite, se toma `TIME`. La forma abreviada es `Tname(variable)`. Ejemplos:

```
TABLE UIC(VC1)
TREGEL(PU4)
```

`EQUATION name(arg1, arg2,...).` Esta forma sirve para calcular magnitudes de la red mediante una instrucción aritmética. Actúa como una llamada a función que recibe los argumentos `arg1`, `arg2`, etc. La fórmula se describe bajo `FUNCTIONS`. Los argumentos pueden ser `variable`, `constant` o `TABLE...`. Resulta útil cuando la misma regla debe aplicarse varias veces a diferentes valores. Su forma abreviada es `Qname(arg1, arg2,...)`. Ejemplos:

```
EQUATION A1 (E1, TIME,23.4,TABLE9)
QA1(220.,TIME,RA,T9(VRA))
```

`EXPRESSION name(instrucción aritmética).` Tiene el mismo efecto que `EQUATION`, pero permite escribir directamente el cálculo; conviene cuando se evalúa una magnitud una sola vez. Para evitar confusiones con parámetros internos de SCEPTRE, es preferible que `name` esté formado solamente por dígitos. La fórmula usa la notación de FORTRAN: además de las cinco operaciones básicas (`+`, `-`, `/`, `*`, `**`), admite funciones externas. Las constantes numéricas deben incluir un punto decimal. La forma abreviada es `Xname(...)`. Ejemplos:

```
E1=X100(310.*DSIN(314.*TIME))
POM=X209(2.*PI*TIME)
PI=X210(4.*DATAN(1.))
CBE=X2(IR5*XTABLE(T23,VRA))
```

`DIODE TABLE name(variable).` Si la característica de un diodo se da como tabla, `variable` se refiere automáticamente a la tensión del diodo. Ejemplo: `JA=DIODE TABLE OY100 (VCJA).`
`DIODE EQUATION (x1, x2).` Una característica de diodo también puede expresarse como

$$J = x_1(e^{x_2 V_J} - 1).$$

SCEPTRE sustituye automáticamente el diodo por una fuente de corriente `J` controlada por esta fórmula; V_J es la tensión del diodo. Ejemplo: `DIODE EQUATION (3.E-14,38.).`
`external function.` Las instrucciones aritméticas pueden llamar a las funciones de la biblioteca de subprogramas FORTRAN y a subprogramas propios. Deben usarse las variantes de doble precisión; por ejemplo, `DSIN`, `DCOS`, `DEXP`, `DATAN` y `DLOG10` para argumentos reales. En el análisis AC se dispone de las funciones para argumentos complejos `ZSQRT`, `ZSIN`, `ZCOS`, `DZABS`, `ZLOG` y `ZCMPLX`, con una grafía ligeramente modificada.
`Parámetros internos de SCEPTRE.` Son variables que el programa necesita y que el usuario puede emplear directamente en las instrucciones de entrada o, por medio de un área `COMMON`, en sus subprogramas FORTRAN. Ejemplos: `XSTPSZ`, `XIR`, `XTMON`, `XSTOPT`.

2.2. Preparación

Primero se dibuja el circuito equivalente de la red que se quiere estudiar y se asigna un nombre a cada elemento. La primera letra del nombre clasifica al elemento según la tabla 2.1.

Tabla 2.1: Clasificación de los elementos

Primera letra	Elemento
R	Resistencia
C	Capacitancia
L	Inductancia
M	Inductancia mutua
E	Fuente de tensión
J	Fuente de corriente

También se da un nombre a todos los nodos. Los nombres solo pueden contener caracteres alfanuméricos y su longitud está limitada como indica la tabla 2.2. Los nombres marcados con ‡ pueden comenzar por una cifra, como el modelo `2N3055` o el nodo `0`. Los límites marcados con † son recomendaciones que conviene no superar.[1]

Tabla 2.2: Longitud máxima de los nombres elegidos por el usuario

Nombre	Circuito	Modelo
Nodos ‡	6	3†
Elementos	5	2†
Parámetros definidos	6	3†
Tablas	5	2†
`EQUATION`	5	2†
Modelos ‡	18	18
Variables de salida	6	3†
Llamadas a modelos	3†	—

Las unidades pueden elegirse libremente, pero deben satisfacer de manera coherente las relaciones

$$U = IR, \qquad \tau = \frac{L}{R} = RC, \qquad f = \frac{1}{T}.$$

La tabla 2.3 muestra ejemplos de sistemas coherentes.

Tabla 2.3: Ejemplos de sistemas de unidades coherentes

Magnitud	Sistema 1	Sistema 2	Sistema 3	Sistema 4
Resistencia	Ω	kΩ	kΩ	kΩ
Capacitancia	F	pF	μF	nF
Inductancia	H	μH	H	mH
Corriente	A	mA	A	mA
Tensión	V	V	kV	V
Tiempo	s	ns	ms	μs
Frecuencia	Hz	GHz	kHz	MHz

Además de los elementos de la red, SCEPTRE permite definir tablas, instrucciones aritméticas y otras magnitudes. También se identifican por su primera letra, según la tabla 2.4.

[1] Cuando se insertan modelos, el nombre de un elemento en el circuito completo, cuya longitud máxima es cinco caracteres, se forma combinando el nombre de la llamada al modelo con el nombre del elemento dentro del modelo.

Tabla 2.4: Clasificación de otras magnitudes de SCEPTRE

Primera letra o palabra	Significado
`P`	Parámetro definido, real
`W`	Parámetro definido, complejo
`T`, `TABLE`	Tabla
`Q`, `EQUATION`	Fórmula llamada por su nombre
`X`, `EXPRESSION`	Expresión aritmética directa
`V`	Tensión en un elemento de la red
`I`	Corriente por un elemento de la red
`D`	Derivada diferencial respecto del tiempo
`G`	Derivada total
`F`	Subprograma FORTRAN propio

Se fija un sentido de circulación de la corriente en cada rama y una polaridad para cada fuente de tensión. SCEPTRE toma las flechas de tensión y corriente con el mismo sentido en los elementos consumidores y con sentidos opuestos en las fuentes.

2.3. Formato de entrada

Todas las instrucciones de SCEPTRE deben escribirse en mayúsculas, entre las columnas 1 y 72 de un archivo de texto. El formato de entrada es libre: se pueden agregar espacios y líneas en blanco para facilitar la lectura. En una misma línea se admiten varias instrucciones separadas por comas.

Una instrucción puede continuar en la línea siguiente si el corte ocurre inmediatamente después de uno de estos caracteres:

```
+   -   *   /   ,   .   (   )   =
```

SCEPTRE ignora lo que aparece después de un punto y coma (`;`) dentro de una línea. El archivo no debe contener caracteres especiales como el tabulador (`TAB`).

2.4. Estructura de la entrada

La tabla 2.5 presenta la estructura de los datos de entrada de SCEPTRE. Dentro de un mismo nivel de instrucciones, el orden es libre. Cada instrucción de control va seguida de los datos que le corresponden y que se describen en los apartados siguientes. Si una instrucción de control no tiene datos, también puede omitirse.

Tabla 2.5: Estructura de los datos de entrada de SCEPTRE

Encabezado principal	Primer nivel	Segundo nivel
CIRCUIT DESCRIPTION	Hasta 11 líneas de comentario; ELEMENTS; DEFINED PARAMETERS; OUTPUTS; FUNCTIONS; INITIAL CONDITIONS; RUN CONTROLS	Bajo RUN CONTROLS: SENSITIVITY; MONTE CARLO; WORST CASE; OPTIMIZATION
MODEL DESCRIPTION	MODEL name	Hasta 11 líneas de comentario; ELEMENTS; DEFINED PARAMETERS; OUTPUTS; FUNCTIONS; INITIAL CONDITIONS
RERUN DESCRIPTION	Hasta 11 líneas de comentario; ELEMENTS; DEFINED PARAMETERS; INITIAL CONDITIONS; FUNCTIONS; RUN CONTROLS	
CONTINUE	RUN CONTROLS	
RE-OUTPUT	OUTPUTS; RUN CONTROLS	
END		

2.5. ELEMENTS

Todas las componentes de la red se definen bajo este encabezado. Su forma general es

```
element, node1 - node2 = value
```

Con ello se establece que la corriente del elemento circula desde `node1` hacia `node2`.

2.5.1. Forma general de los elementos de dos terminales

Para resistencias, condensadores, inductancias, fuentes de tensión y fuentes de corriente se usa la forma siguiente; las convenciones de corriente y tensión se muestran en la tabla 2.1:

```
{R | C | L | E | J} name, node1 - node2 = value
```

Las barras verticales de esta representación indican alternativas para la primera letra del nombre; no se escriben literalmente en el archivo de entrada.

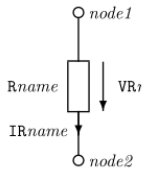
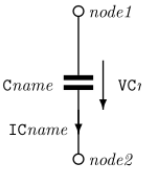
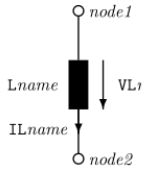
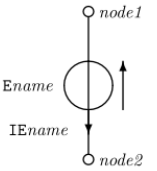
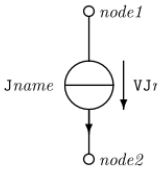
	*Rname , node1 - node2 = value*
	*Cname , node1 - node2 = value*
	*Lname , node1 - node2 = value*
	*Ename , node1 - node2 = value*
	*Jname , node1 - node2 = value*

Figura 2.1: Sentidos de corriente y tensión para los elementos generales de dos terminales. Diagramas reproducidos del manual original.

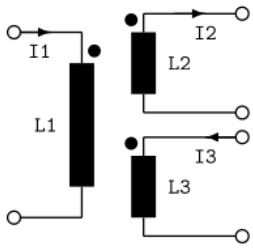
2.5.2. Inductancias mutuas

```
M name, L name1 - L name2 = value
```

`Lname1` y `Lname2` son las dos inductancias acopladas. Debe cumplirse $M < \sqrt{L_1 L_2}$.

El signo de M es positivo cuando las tensiones inducidas por la inductancia mutua y la autoinductancia tienen el mismo sentido en la disposición de los devanados. Para el esquema del manual original se obtiene:

```
M12, L1 - L2 = - value
M13, L1 - L3 = + value
M23, L2 - L3 = - value
```



2.5.3. Fuentes dependientes de resistencias

```
E name, node1 - node2 = constant * VR...
J name, node1 - node2 = constant * IR...
```

2.5.4. Fuentes de corriente controladas

Las fuentes de corriente controladas sirven principalmente para simular circuitos de semiconductores.

Fuentes de primer orden

```
J name, node1 - node2 = {DIODE TABLE... | DIODE EQUATION...}
```

Fuentes de segundo orden

```
J name1, node1 - node2 = value * J name2
```

`Jname2` es una fuente de corriente de primer orden.

2.5.5. Derivadas temporales de las fuentes

Si la red contiene una malla formada únicamente por una fuente de tensión y un condensador, o una fuente de corriente y una inductancia, SCEPTRE necesita además la derivada temporal de esa fuente.[2]

```
{DE | DJ} name = value
```

2.5.6. Llamada a un modelo

```
name, node1 - node2 - ... = MODEL modelname
```

2.5.7. Elementos con límites de tolerancia

Estos datos solo se necesitan para `MONTE CARLO`, `WORST CASE` u `OPTIMIZATION`. Se admite una de las dos formas:

```
{R | E | J} name, node1 - node2 = number1 (number2, number3)
{R | E | J} name, node1 - node2 = number1 (number4)
```

`number1` es el valor nominal; `number2` y `number3`, los límites inferior y superior; `number4`, la desviación porcentual.

[2] El caso puede evitarse colocando una resistencia adecuada en esas mallas.

2.5.8. Fuentes en el análisis de corriente alterna

Una fuente de tensión (`E`) o de corriente (`J`) puede definirse para el análisis AC mediante dos argumentos y una indicación de cómo interpretarlos:

```
{E | J} name, node1 - node2 =
    (argument1, argument2), {DEGREES | RADIANS | COMPLEX}
```

Cada argumento puede ser una constante, un parámetro definido, una tabla o `FREQ`. Con `DEGREES` o `RADIANS`, el primer argumento es la amplitud y el segundo es el ángulo de fase, expresado en grados o radianes respectivamente. Con `COMPLEX`, los argumentos son la parte real y la parte imaginaria.

En el análisis AC, SCEPTRE considera constantes todos los valores de `R`, `C`, `L` y `M`. Emplea automáticamente las condiciones iniciales dadas por el usuario o, si no se especificaron, las calcula para $t = 0$. Las fuentes constantes y las que dependen del tiempo se sustituyen por el valor cero. A diferencia del análisis transitorio, en ciertas topologías no es necesario indicar explícitamente la derivada temporal de una fuente: se obtiene multiplicando por $\mathrm{j}\omega$.

2.6. DEFINED PARAMETERS

Bajo este encabezado pueden definirse variables intermedias adicionales; por ejemplo, para calcular una magnitud nueva a partir de varias magnitudes de la red. La forma general es:

```
Pname = value
```

Ejemplos:

```
PR5 = X10(IR5 * VR5)
PTAU = X11(C1 * R9)
```

También puede indicarse la derivada temporal de una variable, $\mathrm{d}P(t)/\mathrm{d}t$:

```
DPname = value
```

El siguiente ejemplo resuelve un sistema de ecuaciones diferenciales de primer orden:

$$\begin{aligned} \dot{x} &= -6x + 5y + 10, \\ \dot{y} &= 5x - 7y + 0{,}2z, \\ \dot{z} &= 0{,}2y - 0{,}2z - 0{,}5, \end{aligned}$$

con condiciones iniciales $x(0) = 6$, $y(0) = 5$ y $z(0) = 4$. Se escribe en SCEPTRE:

```
DPX=X1(-6.*PX+5.*PY+10.)
DPY=X2(5.*PX-7.*PY+0.2*PZ)
DPZ=X3(0.2*PY-0.2*PZ-0.5)
PX=6
PY=5
PZ=4
```

Las variables intermedias también pueden tener límites de tolerancia, como los elementos de la sección 2.5:

```
Pname = number1 (number2, number3)
Pname = number1 (number4)
```

El diferencial total de una función $P(x, y, \ldots)$ se necesita en los análisis de sensibilidad, optimización y peor caso. Su expresión es

$$\mathrm{d}P = \frac{\partial P}{\partial x}\,\mathrm{d}x + \frac{\partial P}{\partial y}\,\mathrm{d}y + \cdots$$

y se define en SCEPTRE de la siguiente manera:

```
GPname = Pname1 * Dname2 + Pname3 * Dname4 + ...
```

`Pname1` y `Pname3` representan las derivadas parciales $\partial P/\partial x$ y $\partial P/\partial y$; `Dname2` y `Dname4`, los diferenciales $\mathrm{d}x$ y $\mathrm{d}y$. El diferencial total debe escribirse precisamente así: no se admite para él una instrucción `EXPRESSION` o `EQUATION`. Ejemplo:

```
PWR = X1(IRL ** 2. + ILX ** 2.)
GPWR= P2 * DIRL + P3 * DILX
P2 = X76(2. * IRL)
P3 = X77(2. * ILX)
```

Hay que indicar los diferenciales `Dname2`, `Dname4`, etcétera, cuando las variables `name2`, `name4`, etcétera, se definen como independientes.

En el análisis AC, las variables intermedias deben ser complejas. Se les pueden asignar valores reales o complejos, respectivamente:

```
Wname = value
Wname = (value1, value2) typ
```

El significado de los argumentos de `typ` es el indicado para las fuentes AC en la sección 2.5.8. Las variables intermedias complejas solo pueden emplearse para calcular magnitudes de salida: no deben aparecer en una instrucción `EQUATION`, `EXPRESSION` o `TABLE`, ni dentro de una función.

2.7. OUTPUTS

Bajo este encabezado se enumeran las magnitudes de tipo variable que se desea obtener. SCEPTRE puede imprimirlas o graficarlas en función del tiempo, de la frecuencia o de otras variables independientes. Una salida tabulada en función del tiempo o de la frecuencia se solicita así:

```
variable1 [, variable2...]
```

Para obtener además un gráfico de impresora de línea, una variable por página, se agrega `PLOT`:

```
variable1 [, variable2...], PLOT
```

Un gráfico en función de otra variable independiente se solicita mediante:

```
variable1 [, variable2...], PLOT (variable3)
```

Todas las magnitudes excepto `TIME` pueden cambiar de nombre en la salida:

```
variable1 (name1) [, variable2 (name2) ...],
    PLOT (variable3 (name3))
```

Los nombres nuevos pueden tener como máximo seis caracteres. Ejemplos del manual:

```
E1, IE1, VRL, XSTPSZ, PLOT
E1(U1),IE1(I1),VRL(U2),IRL(I2),PLOT
VRL(U2), PLOT (IRL(I2))
```

En los casos anteriores, cada variable se dibuja por separado en una página. También pueden dibujarse juntas hasta nueve magnitudes en un mismo gráfico, distribuido en varias páginas, pero únicamente en función de `TIME`; para ello se asigna un nombre al gráfico, de hasta seis caracteres alfanuméricos:

```
variable1 [, variable2...], PLOT plotname
```

Pueden pedirse varios gráficos de ese tipo con nombres distintos, y también cambiarse los nombres de sus variables de salida. Como uno de esos gráficos puede alcanzar 2000 líneas, debe incluirse bajo `RUN CONTROLS` la instrucción:

```
PLOT INTERVAL = number
```

Los resultados del análisis AC suelen presentarse en función de la frecuencia. Para escoger su representación se escribe:

```
variable1 [, variable2...], DEGREES [, PLOT]
variable1 [, variable2...], RADIANS [, PLOT]
variable1 [, variable2...], COMPLEX [, PLOT]
variable1 [, variable2...], NYQUIST [, PLOT]
```

`NYQUIST` traza la parte imaginaria frente a la real, y permite omitir la palabra `PLOT`. `DEGREES` produce amplitud y ángulo de fase en grados; `RADIANS`, amplitud y ángulo en radianes; `COMPLEX`, partes real e imaginaria.

2.8. INITIAL CONDITIONS

Pueden fijarse condiciones iniciales para condensadores, inductancias y fuentes de corriente de diodo:

```
VCname = number
ILname = number
VJname = number
```

SCEPTRE ofrece dos formas de establecerlas:

1. Solicitar el análisis de corriente continua con `RUN IC`. Entonces el programa calcula las condiciones iniciales, y los valores especificados como arriba sirven solamente como semillas de la iteración de Newton–Raphson. Si no se especifican, la iteración comienza desde cero.
2. Introducir directamente los valores iniciales. Los que no se indiquen toman automáticamente el valor cero.

2.9. FUNCTIONS

Bajo este encabezado se describen todas las tablas y funciones que necesitan las secciones ELEMENTS y DEFINED PARAMETERS.

2.9.1. Instrucción EQUATION

La forma general es:

```
EQUATION name (list) = (arithmetic expression)
Q name (list) = (arithmetic expression)
```

La lista contiene variables auxiliares o parámetros formales. Sus nombres tienen como máximo seis caracteres y no pueden comenzar por una cifra ni por una de las letras I a N. La expresión aritmética las combina con la sintaxis de las funciones de fórmula de FORTRAN; también puede llamar a subprogramas externos. Todas las constantes numéricas deben escribirse con punto decimal. Para operaciones complejas, los nombres de los parámetros auxiliares deben comenzar por Z. El número de parámetros formales debe coincidir con el de argumentos de cada llamada a EQUATION. Los nombres de parámetros formales pueden reutilizarse en otras instrucciones EQUATION.

Ejemplo con una fuente definida bajo ELEMENTS:

```
ELEMENTS
  E1, 1-0 = Q9 (220.,50.)
  ...
FUNCTIONS
  Q9 (A,B) = (A*DSQRT(2.) * DSIN (B*6.28 * TIME))
```

Ejemplo con un parámetro definido:

```
DEFINED PARAMETERS
  WZE = Q107 (EIN,IEIN)
  ...
FUNCTIONS
  Q107 (Z1,ZB7X) = (Z1/ZB7X)
```

2.9.2. Instrucción TABLE

Una tabla se declara con TABLE o su abreviatura T; para una tabla de diodo se antepone DIODE. Luego se indica el nombre y los valores, separados por comas:

```
[DIODE] TABLE name
number1, number2, number3, number4, ...
```

Los valores también pueden comenzar después de = o de una coma en la misma línea de la declaración, en vez de empezar en la línea siguiente.

Los valores se introducen por pares: primero la variable independiente y luego la dependiente. Los valores de la variable independiente deben estar ordenados monótonamente en forma ascendente, y la función debe estar unívocamente definida. Para representar una discontinuidad pueden darse dos valores dependientes distintos para un mismo valor independiente, como en el ejemplo. Los valores intermedios se interpolan o extrapolan linealmente; para extrapolar importa especialmente el último par de valores.

La tabla siguiente describe la función de tensión de la figura 2.2:

```
TABLE XYZ7
0,0, 0,1, 2,1, 2,-2, 4,-2, 4,3,
6,3, 6,0.5, 100,0.5
```

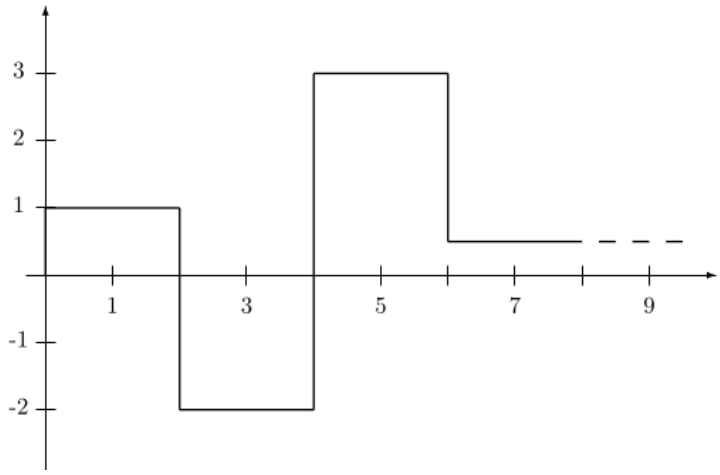


Figura 2.2: Función de tensión escalonada del ejemplo `TABLE XYZ7`. Gráfico reproducido del manual original.

Si la tabla se declara como `DIODE TABLE`, SCEPTRE interpreta el primer valor de cada par como la tensión del diodo y el segundo como la corriente correspondiente.

2.10. RUN CONTROLS

Este apartado reúne las instrucciones que gobiernan el cálculo. Hay tres tipos de análisis, algunos de los cuales pueden combinarse.

2.10.1. Análisis de corriente continua (DC)

Para ejecutar solamente el análisis de corriente continua basta con:

```
RUN INITIAL CONDITIONS ONLY
```

En este caso se emplea el algoritmo de Newton–Raphson. El límite predeterminado es de 100 iteraciones; también pueden modificarse los criterios de convergencia:

Instrucción	**Valor predeterminado**
`NEWTON-RAPHSON PASS LIMIT = number`	100
`RELATIVE CONVERGENCE = number`	10^{-3}
`ABSOLUTE CONVERGENCE = number`	10^{-4}

La instrucción siguiente llama a un algoritmo implícito en lugar de Newton–Raphson:

```
RUN IC VIA IMPLICIT
```

Dentro del análisis DC hay cuatro procedimientos especiales. Además de sus instrucciones en RUN CONTROLS, necesitan declaraciones bajo CIRCUIT DESCRIPTION. Para MONTE CARLO, WORST CASE y OPTIMIZATION debe declararse al menos una magnitud con límites de tolerancia bajo ELEMENTS o DEFINED PARAMETERS. Cuando se piden varios procedimientos, se ejecutan siempre en este orden, independientemente del orden de las instrucciones:

1. SENSITIVITY;
2. MONTE CARLO;
3. WORST CASE;
4. OPTIMIZATION.

Sensibilidad. Se solicita con:

```
RUN SENSITIVITY
```

Monte Carlo. La forma general es RUN MONTE CARLO [= number]; number indica el número de iteraciones y vale 10 por omisión. La distribución y la semilla se controlan con:

```
DISTRIBUTION = GAUSSIAN
DISTRIBUTION = UNIFORM
INITIAL RANDOM NUMBER = number
```

El valor predeterminado de INITIAL RANDOM NUMBER es 127263527. Si se especifica, debe ser un entero positivo, impar y de nueve cifras. Para comenzar siempre con la misma semilla en las repeticiones se usa INITIAL RANDOM NUMBER = DEFAULT. La instrucción LIST MONTE CARLO DETAILS produce información adicional de cada iteración.
Peor caso. La forma general es:

```
RUN WORST CASE [= LOW]
RUN WORST CASE [= NOMINAL]
RUN WORST CASE [= HIGH]
```

Las palabras después de = indican qué valores se usarán como condiciones iniciales de un eventual análisis transitorio posterior.
Optimización. Se solicita con RUN OPTIMIZATION [= number]; la cantidad predeterminada de iteraciones es 30. LIST OPTIMIZATION DETAILS muestra resultados intermedios. Las instrucciones de control adicionales y sus valores predeterminados son:

Instrucción	Valor predeterminado
INITIAL H MATRIX FACTOR = number	1
OPTIMIZATION CRITERION = number	10^{-7}
OPTIMIZATION RANDOM STEPS = number	0
RANDOM STEP SIZE CONTROL = number	0,2
MINIMUM FUNCTION ESTIMATE = number	0

También puede usarse PUNCH OPTIMIZATION RESULTS. El significado y la utilización detallada de estas instrucciones se explican en la referencia [3] del manual original.

2.10.2. Análisis de corriente alterna (AC)

Se inicia con `RUN AC`. Para calcular una sola frecuencia basta con `FREQUENCY = number`. Para describir un intervalo de frecuencias se emplean las instrucciones siguientes:

Instrucción	**Valor predeterminado**
`INITIAL FREQUENCY = number`	–
`FINAL FREQUENCY = number`	–
`NUMBER FREQUENCY STEPS = number`	10
`TYPE FREQUENCY RUN = LINEAR`	–
`TYPE FREQUENCY RUN = LOG`	–

Si en las repeticiones solo cambian las fuentes de corriente o de tensión, la siguiente instrucción ahorra tiempo de cálculo:

```
USE FIXED AC MATRIX IN RERUNS
```

2.10.3. Análisis transitorio (TR)

Para el análisis transitorio debe indicarse siempre el tiempo final:

```
STOP TIME = number
```

El número máximo de pasos de integración es 20000 por omisión y el tiempo inicial es 0; ambos pueden modificarse:

```
MAXIMUM INTEGRATION PASSES = number
START TIME = number
```

El método de integración se elige con una de estas variantes:

```
INTEGRATION ROUTINE = XPO
INTEGRATION ROUTINE = TRAP
INTEGRATION ROUTINE = RUK
INTEGRATION ROUTINE = IMPLICIT
```

Para el método implícito puede indicarse cómo calcular el jacobiano:

```
USE DIFFERENCED JACOBIAN
USE SYMBOLIC JACOBIAN
```

Si esta última instrucción falta, SCEPTRE elige automáticamente el algoritmo que considera más adecuado para la red.

SCEPTRE recalcula un tamaño de paso variable después de cada paso de integración. Las tres instrucciones de control y sus valores predeterminados se muestran en la tabla 2.6. *Todos los valores predeterminados de esa tabla se multiplican por `STOP TIME`.*

Instrucción	**XPO**	**TRAP**	**RUK**	**IMPLICIT**
`MINIMUM STEP SIZE = number`	10^{-5}	10^{-5}	10^{-5}	10^{-14}
`MAXIMUM STEP SIZE = number`	$2 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	$2 \cdot 10^{-2}$
`STARTING STEP SIZE = number`	10^{-3}	10^{-3}	10^{-3}	10^{-8}

Tabla 2.6: Valores predeterminados de los tamaños de paso, expresados como factores de `STOP TIME`.

La precisión también puede modificarse mediante las cuatro instrucciones de la tabla 2.7; la referencia [3] del manual original explica el significado de estos parámetros. El guion indica que el original no da un valor para ese método.

Instrucción	XPO	TRAP	RUK	IMPLICIT
`MINIMUM ABSOLUTE ERROR = number`	10^{-4}	$5 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	10^{-4}
`MAXIMUM ABSOLUTE ERROR = number`	$5 \cdot 10^{-3}$	10^{-3}	$5 \cdot 10^{-3}$	–
`MINIMUM RELATIVE ERROR = number`	$2 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$5 \cdot 10^{-5}$	–
`MAXIMUM RELATIVE ERROR = number`	$5 \cdot 10^{-3}$	10^{-2}	$5 \cdot 10^{-3}$	–

Tabla 2.7: Valores predeterminados de los controles de error de integración.

2.10.4. Análisis DC seguido de TR

Con `RUN INITIAL CONDITIONS` se hace primero un análisis DC y luego uno transitorio. También debe indicarse `STOP TIME`. Los resultados del *último* análisis DC proporcionan las condiciones iniciales del análisis TR. Pueden combinarse todas las instrucciones de control DC y TR; si se solicitan análisis de tolerancias, estos mantienen el orden indicado en el apartado DC.

2.10.5. Análisis DC seguido de AC

Se aplican las mismas reglas de combinación que para DC seguido de TR: pueden utilizarse todas las instrucciones de control DC y AC.

2.10.6. Análisis AC seguido de TR

Esta combinación no está disponible.

2.10.7. Control de la salida

Los resultados pedidos en `OUTPUTS` están disponibles después de cada paso de integración. `MAXIMUM PRINT POINTS = number` limita cuántos se imprimen; su valor predeterminado es 1000 y con `number = 0` se suprime la salida impresa. `PRINT INTERVAL = number` permite además imprimir resultados a intervalos de tiempo determinados.

Los gráficos combinados se piden bajo `OUTPUTS`. `PLOT INTERVAL = number` controla su longitud en líneas, calculada como el cociente `STOP TIME / PLOT INTERVAL`; el máximo es de 2000 líneas. Si falta esa instrucción, se produce un gráfico individual sencillo para cada magnitud de salida.

2.10.8. Ejecuciones repetidas

En cada repetición se recalculan normalmente las condiciones iniciales, aunque también pueden especificarse explícitamente para cada ejecución. Para usar en cambio los resultados de la primera ejecución o de la inmediatamente anterior se indica, respectivamente:

```
IC FOR RERUNS = MASTER RESULTS
IC FOR RERUNS = PRECEDING RESULTS
```

Las ejecuciones repetidas se describen en la sección 2.13 del manual original.

2.10.9. Control de la ejecución

Puede interrumpirse una ejecución en función del comportamiento de la red mediante:

```
TERMINATE IF (variable .ro. variable)
STOP IF (variable .ro. constant)
```

En cualquiera de los dos lados de la comparación puede usarse una variable o una constante. `.ro.` representa `.LE.`, `.LT.`, `.GT.`, `.GE.`, `.EQ.` o `.NE.`, con el significado habitual de FORTRAN. Las condiciones pueden combinarse mediante `AND` y `OR`. `TERMINATE` acaba la ejecución actual, pero deja que se realicen las repeticiones pendientes; `STOP` concluye todo el cálculo. Ejemplos:

```
TERMINATE IF (PWR .GT. 0.3)
STOP IF ((VRL .LE. E3) .OR. (IL9 .GT. 20.))
```

Durante el cálculo, SCEPTRE guarda periódicamente datos internos que permiten reanudarlo tras una interrupción anormal. El intervalo predeterminado es de 15 minutos y puede modificarse con `COMPUTER SAVE INTERVAL = number`. La reanudación se describe en la sección 2.14 del original.

También se pueden limitar el tiempo de CPU y el tiempo transcurrido mientras se ejecuta el programa. `SOLUTION TIME LIMIT = number` fija los segundos de CPU; solo cuenta la fase de ejecución. `COMPUTER TIME LIMIT = number` fija los minutos transcurridos y vale 600 por omisión. Al alcanzarse cualquiera de estos límites, SCEPTRE entrega los resultados calculados hasta ese momento y finaliza ordenadamente.

Si solo se desea procesar las instrucciones SCEPTRE, por ejemplo para comprobar su sintaxis, sin ejecutar la simulación, se utiliza:

```
EXECUTE SETUP PHASE ONLY
```

2.10.10. Ayudas para las pruebas

Instrucción	Información producida
`LIST NODE MAP`	nodos de la red y ramas correspondientes
`WRITE SIMUL8 DATA`	programas FORTRAN generados por SCEPTRE
`PRINT B MATRIX`	matriz topológica de la red
`PRINT A MATRIX`	matriz A del análisis AC
`PRINT EIGENVALUES`	autovalores del análisis AC
`PRINT EIGENVECTORS`	autovectores del análisis AC

Las tres últimas instrucciones de la tabla se aplican al análisis AC.

2.11. SENSITIVITY, MONTE CARLO, WORST CASE y OPTIMIZATION

Estos análisis de tolerancias se declaran bajo encabezados del mismo nombre:

```
SENSITIVITY
MONTE CARLO
WORST CASE
OPTIMIZATION
```

A cada encabezado le sigue al menos una lista de la forma:

```
(dependent_variable1, ... / independent_variable1, ...)
```

Las variables dependientes definen las magnitudes objetivo; dentro de cada grupo, los nombres se separan con comas. Debe haber al menos una variable dependiente y una independiente. Para `MONTE CARLO`, `WORST CASE` y `OPTIMIZATION`, las independientes deben declararse con límites de tolerancia. Las tablas 2.8 y 2.9 muestran los tipos permitidos.

Variable	Magnitud
`VCname`	tensión de un condensador
`IEname`	corriente de una fuente de tensión independiente
`Jname`	fuente de corriente dependiente de primer orden
`VJname`	tensión de una fuente de corriente independiente
`ILname`	corriente de una inductancia
`VRname`	tensión de una resistencia
`IRname`	corriente de una resistencia
`Pname`	parámetro definido

Tabla 2.8: Tipos de variables dependientes admitidos por los análisis de tolerancias.

Variable	Magnitud
`Rname`	resistencia
`Ename`	fuente de tensión independiente
`Jname`	fuente de corriente independiente
`Pname`	parámetro definido

Tabla 2.9: Tipos de variables independientes admitidos por los análisis de tolerancias.

Si `Pname` es dependiente en un análisis de sensibilidad, optimización o peor caso, también debe especificarse su diferencial total `GPname`. Como variable independiente, `Pname` solo puede aparecer como factor de una fuente de corriente de segundo orden; por ejemplo, `Jname1 = Pname * Jname2`.

Ejemplo de sensibilidad:

```
SENSITIVITY
  (IL3, PX / IX, EY)
  (VR1, IR1, VCX / P1, E1)
```

La primera lista pide $\partial IL3/\partial IX$, $\partial IL3/\partial EY$, $\partial PX/\partial IX$ y $\partial PX/\partial EY$. La segunda pide las seis derivadas que resultan de combinar `VR1`, `IR1` y `VCX` con `P1` y `E1`.

Ejemplo de optimización:

```
OPTIMIZATION
  (IR1, P1 / R1, P2, E1)
  (VC1 / I1, R2)
```

`IR1` y `P1` se optimizan respecto de `R1`, `P2` y `E1`; `VC1`, respecto de `I1` y `R2`.

2.12. Modelos

Las partes de una red o los circuitos equivalentes usados con frecuencia pueden almacenarse temporal o permanentemente e insertarse después en la red estudiada. Al insertarlos

también pueden cambiarse parámetros concretos. La descripción precede a la de la red y comienza con:

```
MODEL DESCRIPTION [( [INITIAL,] PRINT )]
```

Si no se emplean modelos, esta sección se omite. `INITIAL` solo se necesita cuando se desea almacenar permanentemente el primer modelo; `PRINT` hace imprimir su descripción. Ambas opciones se admiten únicamente para modelos permanentes.

2.12.1. Descripción de un modelo

La declaración admite la siguiente forma:

```
MODEL modelname [(PERM)] (node1 - node2 - ...)
MODEL modelname [(TEMP)] (node1 - node2 - ...)
```

`PERM` indica almacenamiento permanente y `TEMP`, eliminación al terminar la simulación. El nombre del modelo puede tener hasta 18 caracteres y el modelo hasta 25 puntos de conexión. Un modelo permanente se elimina con:

```
MODEL modelname (DELETE)
```

Después de `MODEL` puede haber hasta once líneas de comentarios, seguidas de instrucciones bajo `ELEMENTS`, `DEFINED PARAMETERS`, `OUTPUTS`, `FUNCTIONS` o `INITIAL CONDITIONS`. Rigen las mismas reglas que para las instrucciones correspondientes de `CIRCUIT DESCRIPTION`. Ejemplo del original:

```
MODEL DESCRIPTION (INITIAL,PERM)
MODEL 2N1734B (PERM) (B-E-C)
  ELEMENTS
    CE, B-E = Q1 (5., 80., TABLE 1 (VCE))
    CC, B-C = Q1 (10., 200., TABLE 2 (VCC))
    J1, B-E = DIODE TABLE 1
    JA, E-B = 0.1 * J2
    J2, B-C = DIODE TABLE 2
    JB, C-B = 0.98 * J1
  OUTPUTS
    VCE, VCC, J1, PLOT
  FUNCTIONS
    DIODE TABLE 1 = 0,0, .3,0, .65,.5, ....
    DIODE TABLE 2 = 0,0, .58,0, .62,.4, ....
    EQUATION 1 (A,B,C) = (A + B * C)
```

Incluir `OUTPUTS` dentro del modelo es una decisión que depende de cada caso. El ejemplo escribe `PERM` en `MODEL DESCRIPTION`, aunque la forma general impresa arriba muestra `PRINT`; se conserva la discrepancia del original.

2.12.2. Inserción de un modelo

Bajo `CIRCUIT DESCRIPTION/ELEMENTS` se inserta un modelo con su nombre de referencia, nodos de conexión y nombre de modelo. Dentro de los paréntesis pueden indicarse su clase de almacenamiento, modificaciones, salidas suprimidas y la petición de imprimir su descripción:

```
name, node1 - node2 - ... = MODEL modelname
    (PERM|TEMP [, CHANGE ...] [, SUPPRESS ...] [, PRINT])
```

Las barras y los corchetes de esta forma esquemática indican alternativas y partes opcionales, no caracteres literales de entrada. `modelname` identifica el modelo definido o almacenado. SCEPTRE añade el nombre de referencia `name` a todos los nombres internos del modelo —elementos, nodos y otras magnitudes— para que sigan siendo únicos aunque se inserte el mismo modelo varias veces. Después lo trata como si se hubiera definido directamente bajo `CIRCUIT DESCRIPTION`. Debido al límite de longitud de nombres descrito en la tabla 2.2 del original, conviene que los nombres internos y el nombre de referencia sean breves.

Ejemplo de inserción del modelo `2N1734B`:

```
T1, 13-4-GND = MODEL 2N1734B (PERM)
```

Una magnitud interna de esa instancia se identifica añadiendo `T1` a su nombre: `CET1`, `CCT1`, `J1T1`, `VCET1`, `ICET1`, etcétera.

`CHANGE` modifica valores de elementos y parámetros definidos del modelo. Los ejemplos siguientes continúan con `2N1734B`:

```
T1, 13-4-GND = MODEL 2N1734B (PERM, CHANGE CC=50)
T1, 13-4-GND = MODEL 2N1734B
    (PERM, CHANGE CC=TABLE 7(VCCT1),
     JA=0.2*J2T1, JB=DIODE TABLE 4)
```

En el segundo ejemplo, `TABLE 7` proporciona los valores de `CC`, `DIODE TABLE 4` la característica de la fuente `JB`, y `JA` recibe otra expresión matemática. Ambas tablas deben definirse bajo `CIRCUIT DESCRIPTION/FUNCTIONS`. También puede sustituirse la expresión de `CC` por una función:

```
T1, 13-4-GND = MODEL 2N1734B
    (PERM, CHANGE CC=EQUATION 5 (VCCT1,VCX))
```

`EQUATION 5` debe definirse bajo `CIRCUIT DESCRIPTION/FUNCTIONS`. Sus argumentos son las tensiones de los condensadores `CC` y `CX`; este último no pertenece al modelo. Es posible reemplazar una tabla por otra:

```
T1, 13-4-GND = MODEL 2N1734B
    (PERM, CHANGE TABLE 1 = TABLE 7)
```

`TABLE 7` también debe definirse bajo `CIRCUIT DESCRIPTION/FUNCTIONS`. Las dos tablas pueden tener distinta cantidad de pares de valores. En cambio, el original advierte que no se puede sustituir directamente una característica de diodo por otra mediante `CHANGE DIODE TABLE 1 = DIODE TABLE 9`.

```
T1, 13-4-GND = MODEL 2N1734B (PERM, CHANGE Q1=Q2)
```

La nueva función `EQUATION 2`, definida bajo `CIRCUIT DESCRIPTION/FUNCTIONS`, reemplaza a `EQUATION 1` y debe tener los mismos parámetros. No es posible reemplazar una función por una tabla ni a la inversa, aunque los valores de elementos y parámetros definidos sí pueden reemplazarse por constantes, tablas o funciones.

`SUPPRESS` evita total o parcialmente la salida pedida bajo `MODEL DESCRIPTION/OUTPUTS`; `PRINT` hace listar la descripción del modelo. Estas opciones pueden combinarse con `CHANGE`:

```
T1, 13-4-GND = MODEL 2N1734B (PERM, SUPPRESS J1)
T1, 13-4-GND = MODEL 2N1734B (PERM, SUPPRESS ALL)
T1,13-4-GND=MODEL 2N1734B
    (PERM,CHANGE CC=50,CE=30,SUPPRESS J1,VCE,PRINT)
```

Es preferible definir las condiciones iniciales bajo `CIRCUIT DESCRIPTION/INITIAL CONDITIONS`; por ejemplo, `VCCT1 = 0.53`.

2.13. RERUN DESCRIPTION

SCEPTRE permite repetir una ejecución terminada con parámetros modificados, indicando solamente los valores que cambian. Las modificaciones se agrupan bajo:

```
RERUN DESCRIPTION [(n)]
```

`n` es el número de ejecuciones modificadas. Si hay una sola repetición, se omite el paréntesis. A diferencia de `CIRCUIT DESCRIPTION`, en `RERUN DESCRIPTION` solo puede modificarse una variable por línea de entrada.

Los valores constantes declarados bajo `ELEMENTS` y `DEFINED PARAMETERS` se sustituyen mediante:

```
elementname = number1, number2, ...
parametername = number1, number2, ...
```

`number1` se aplica en la primera repetición, `number2` en la segunda, y así sucesivamente. Ejemplo:

```
RERUN DESCRIPTION (3)
  ELEMENTS
    RLOAD = 25, 200, 20E3
  DEFINED PARAMETERS
    PSI = 13.4, 13.2, 13.1
```

También pueden redefinirse los límites de tolerancia de un elemento o parámetro. Se da un valor nominal y, opcionalmente, un límite inferior y otro superior o una desviación porcentual:

```
R3 = 4.6, 9.(8.,10.5)
```

En la primera repetición, `R3` vale 4,6 sin límites de tolerancia; en la segunda, su valor nominal es 9, con límites 8 y 10,5. Las condiciones iniciales bajo `INITIAL CONDITIONS` pueden cambiarse de igual forma, usando `VCname`, `ILname` o `VJname` a la izquierda:

```
VCname = number1, number2, ...
ILname = number1, number2, ...
VJname = number1, number2, ...
```

Los valores de las tablas se cambian bajo FUNCTIONS. La tabla nueva debe tener tantos pares como la original. En una instrucción RERUN DESCRIPTION, los valores independientes x de una tabla solo pueden reemplazarse una vez; los valores dependientes y correspondientes pueden cambiarse tantas veces como se necesite. Para N repeticiones se usa:

```
TABLE name = X1, Y11, Y12,...,Y1N, X2, Y21, Y22,...,Y2N,...
```

Ejemplo:

```
CIRCUIT DESCRIPTION
  .....
  FUNCTIONS
    TABLE 7 = 0,0, 1,10, 2,27, 4,47
  .....
RERUN DESCRIPTION (2)
  FUNCTIONS
    TABLE 7 = 0,0,0, 1,3,5, 3,4,8, 5,7,9
```

Los valores resultantes son:

Original		**1.ª repetición**		**2.ª repetición**	
X	Y	X	Y	X	Y
0	0	0	0	0	0
1	10	1	3	1	5
2	27	3	4	3	8
4	47	5	7	5	9

Tabla 2.10: Valores de TABLE 7 en la ejecución original y en dos repeticiones.

En la segunda repetición se reutilizan los valores x de la primera. Para cambiarlos también en la segunda, hace falta otro encabezado RERUN DESCRIPTION:

```
RERUN DESCRIPTION
  FUNCTIONS
    TABLE 7 = 0,0, 1,3, 3,4, 5,7
RERUN DESCRIPTION
  FUNCTIONS
    TABLE 7 = 0,0, 2,4, 3,5, 4,9
```

Restricciones de las repeticiones. Bajo RUN CONTROLS pueden modificarse todas las instrucciones excepto STOP y TERMINATE, pero se aplican las restricciones siguientes:

- No puede cambiarse la topología de la red ni lo declarado bajo OUTPUTS.
- Las constantes deben sustituirse por constantes, las tablas por tablas y las instrucciones EQUATION por instrucciones EQUATION.

- Las instrucciones `EXPRESSION` y `EQUATION` solo pueden modificarse indirectamente cuando sus variables están definidas bajo `DEFINED PARAMETERS`: se cambian esos parámetros. Por ejemplo:

```
CIRCUIT DESCRIPTION
  ELEMENTS
    JE, 1-8 = DIODE EQUATION (PX1,PX2)
    .....
  DEFINED PARAMETERS
    PX1 = ...., PX2 = .....
    .....
RERUN DESCRIPTION
  DEFINED PARAMETERS
    PX1 = ....
    PX2 = ....
```

- Cambiar elementos de la red puede requerir recalcular las condiciones iniciales.
- En general no puede cambiarse el tipo de análisis. Las condiciones iniciales no se recalculan en una repetición aunque se hayan calculado en la ejecución original; para recalcularlas en cada repetición debe volver a indicarse `RUN INITIAL CONDITIONS` bajo `RUN CONTROLS`.
- Las instrucciones que siguen solo son válidas en las repeticiones si se calcularon las condiciones iniciales en la ejecución original:

```
RUN SENSITIVITY
RUN MONTE CARLO
RUN WORST CASE
RUN OPTIMIZATION
RUN INITIAL CONDITIONS
RUN INITIAL CONDITIONS ONLY
```

Instrucciones prohibidas en las repeticiones:

```
NO ELEMENT SORT
PUNCH PROGRAM
WRITE SIMUL8 DATA
VECTOR EQUATIONS
WRITE DEBUG
PRINT B MATRIX
LIST NODE MAP
IC FOR RERUNS
USE DIFFERENCED JACOBIAN
USE SYMBOLIC JACOBIAN
EXECUTE SETUP PHASE ONLY
USE FIXED AC MATRIX IN RERUNS
IMPULSE RESPONSE BUFFER
INPUT FUNCTION BUFFER
```

Instrucciones no heredadas de la ejecución original: deben repetirse expresamente si se necesitan.

```
RUN AC
RUN SENSITIVITY
RUN MONTE CARLO
RUN WORST CASE
RUN OPTIMIZATION
LIST MONTE CARLO DETAILS
LIST OPTIMIZATION DETAILS
PUNCH OPTIMIZATION RESULTS
OPTIMIZATION RANDOM STEPS
PRINT A MATRIX
PRINT EIGENVALUES
MINIMUM FUNCTION ESTIMATE
INITIAL RANDOM NUMBER
```

Una elección de `IC FOR RERUNS = MASTER RESULTS` o `IC FOR RERUNS = PRECEDING RESULTS` en `CIRCUIT DESCRIPTION` entra en conflicto con `RUN INITIAL CONDITIONS`, `RUN INITIAL CONDITIONS ONLY` o `RUN IC VIA IMPLICIT` en `RERUN DESCRIPTION`; la combinación aborta el programa.

Para cambiar un modelo durante una repetición rigen las mismas reglas, pero los nombres de sus elementos deben llevar añadido el nombre de referencia de la instancia. Ejemplo:

```
MODEL DESCRIPTION
  MODEL ABCDEFG (PERM) (1-7-8-9)
    ELEMENTS
      ....
      R1, 1-2 = 1
    DEFINED PARAMETERS
      P1 = 2.7
    FUNCTIONS
      TABLE 1 = ...
CIRCUIT DESCRIPTION
  ELEMENTS
    ...
    T8, 4-9-1-2 = MODEL ABCDEFG
    ...
RERUN DESCRIPTION
  ELEMENTS
    R1T8 = 0.98
  DEFINED PARAMETERS
    P1T8 = 3.1
  FUNCTIONS
    TABLE 1 T8 = ...
```

2.14. CONTINUE

`CONTINUE` permite proseguir una ejecución que terminó normalmente. Solo se aplica al análisis transitorio; si hubo repeticiones, continúa la última. No permite modificar la red. Las únicas instrucciones que pueden cambiarse bajo `RUN CONTROLS` son:

- `STOP TIME`, `COMPUTER TIME LIMIT` y `MAXIMUM INTEGRATION PASSES`;

- `INTEGRATION ROUTINE` y `USE DIFFERENCED JACOBIAN` o `USE SYMBOLIC JACOBIAN`;
- `MINIMUM STEP SIZE`, `MAXIMUM STEP SIZE` y `STARTING STEP SIZE`;
- `MINIMUM ABSOLUTE ERROR`, `MAXIMUM ABSOLUTE ERROR`, `MINIMUM RELATIVE ERROR` y `MAXIMUM RELATIVE ERROR`;
- `MAXIMUM PRINT POINTS`, `PRINT INTERVAL` y `PLOT INTERVAL`.

Ejemplo:

```
CONTINUE
  RUN CONTROLS
    STOP TIME = 200
END
```

2.15. RE-OUTPUT

`RE-OUTPUT` genera nuevas copias de los resultados de una ejecución terminada, sin volver a calcularla. Bajo `OUTPUTS` se indican las variables que se imprimirán o graficarán: solo pueden ser variables pedidas en la ejecución original, aunque allí se hubieran presentado de otra manera. Bajo `RUN CONTROLS` puede modificarse la cantidad de puntos de salida con `MAXIMUM PRINT POINTS`. En el análisis AC, los gráficos solo pueden presentarse en las modalidades de salida solicitadas originalmente: `RADIANS`, `DEGREES` o `COMPLEX`.

```
RE-OUTPUT
  OUTPUTS
    VR1, VR2
    VL1, VCC, PLOT
  RUN CONTROLS
    MAXIMUM PRINT POINTS = 1500
END
```

2.16. END

`END` marca el final de la entrada SCEPTRE y debe ser su última instrucción.

3 Indicaciones generales

Este capítulo reúne recomendaciones de uso de SCEPTRE y explica posibles fuentes de error y sus causas.

3.1. Diagnóstico de errores

SCEPTRE dispone de un diagnóstico amplio. Sus mensajes se agrupan en tres clases:

Mensaje	**Consecuencia**
`warning only`	Error generalmente no crítico, por ejemplo una falta de sintaxis que SCEPTRE puede corregir; la simulación se ejecuta.
`simulation deleted`	El error impide la simulación posterior, pero continúa la comprobación de los datos de entrada.
`execution terminated`	Un error grave interrumpe el diagnóstico.

Además de errores de sintaxis o de estructura de la entrada, se detectan problemas que surgen del conjunto de la red, como un *computational delay* o una topología inadmisible. Normalmente los mensajes permiten identificar la causa, aunque una confusión tipográfica —por ejemplo, escribir el dígito cero en lugar de la letra `O`— puede desencadenar numerosos avisos difíciles de interpretar. Los errores lógicos que solo se manifiestan en los resultados no pueden diagnosticarse de esta manera; en algunas situaciones ayudan las instrucciones de prueba del apartado 2.10.10.

3.2. Algunas reglas básicas

- Conviene escoger las unidades físicas para que los valores extremos no difieran en más de cinco o seis órdenes de magnitud y se mantengan próximos a uno. Una escala muy dispar perjudica la convergencia de los métodos iterativos o incluso la impide.

- Al usar modelos, por ejemplo de una biblioteca permanente, hay que comprobar que empleen las mismas unidades que la red en la que se insertan.

- En redes grandes o complejas es recomendable poner en funcionamiento el circuito por etapas y comprobar por separado los modelos y circuitos equivalentes.

- Es preferible describir las no linealidades mediante fuentes controladas en vez de elementos pasivos `R`, `C` o `L`.

- Para una no linealidad dependiente de tensión conviene usar como control la tensión de un condensador; para una dependiente de corriente, la corriente de una inductancia. Ambas son variables de estado y se recalculan en cada paso de integración, lo que ayuda a evitar el *computational delay* descrito en la sección 3.7.

3.3. Inductancias y capacidades dependientes del tiempo

En el análisis transitorio, el método de resolución da resultados incorrectos si una inductancia o una capacidad dependen del tiempo; véase también [1, cap. 1.7] del original. Si $C = f(t)$, se corrige colocando en paralelo con el condensador una fuente de corriente $J = V_C \, \mathrm{d}C/\mathrm{d}t$. Si $L = f(t)$, se coloca en serie una fuente de tensión $E = I_L \, \mathrm{d}L/\mathrm{d}t$. En cambio, las resistencias dependientes del tiempo, las inductancias dependientes de corriente y las capacidades dependientes de tensión se calculan correctamente.

3.4. Tablas

Cuando una tabla describe una no linealidad, conviene suavizar la curva o añadir puntos de apoyo para evitar quiebres. Son especialmente críticos el primer y el último par, porque SCEPTRE los usa para extrapolar fuera del intervalo. Una tabla que describa un elemento pasivo `R`, `C` o `L` no debe dar nunca cero. En general es preferible una función analítica a una tabla.

Cuando se usa una tabla dentro de una instrucción `EXPRESSION` o `EQUATION`, cambia la sintaxis:

```
..... = X4711 ( ..... XTABLE(Tname,variable) ..... )
```

`Tname` es el nombre de la tabla y el segundo argumento es la variable independiente, por ejemplo `TIME`.

3.5. Instrucciones aritméticas

Al describir elementos o parámetros dependientes mediante expresiones aritméticas hay que verificar el dominio de validez. Las funciones estándar de FORTRAN, como `MIN`, `MAX` y `ABS`, pueden evitar divisiones por cero o desbordamientos. Ejemplos del original:

```
CC, B-5 = X1 (PX/(0.8-MIN(0,VCC))**0.33333)
JD, C-B = X2 (PS*EXP(MIN(88,PQ*VJD))-1)
```

3.6. Restricciones topológicas

Si la red tiene mallas compuestas por fuentes de tensión y capacidades, o nodos a los que solo se llega a través de fuentes de corriente e inductancias, las fuentes variables requieren que se indique su primera derivada temporal.

3.6.1. Análisis DC, AC y TR

Ninguno de los tres análisis admite una malla compuesta exclusivamente por fuentes de tensión (*voltage source loop*) ni un conjunto de corte formado solo por fuentes de corriente (*all-current source cut set*).

3.6.2. Análisis DC

Para calcular condiciones iniciales tampoco se admite una malla formada solo por fuentes de tensión e inductancias, ni un nodo al que solo se acceda a través de capacidades

y fuentes de corriente. Para el segundo caso puede unirse el nodo afectado a tierra mediante una resistencia que no altere apreciablemente el circuito.

Estas restricciones pueden evitarse calculando las condiciones iniciales mediante una integración transitoria implícita en lugar de un análisis DC. La sintaxis es:

```
RUN INITIAL CONDITIONS [ONLY]
RUN IC VIA IMPLICIT
```

El procedimiento también puede servir cuando el análisis DC no converge; véase la sección 3.17 del original.

3.7. «Computational delay»

La posibilidad de describir cada elemento en función de otras magnitudes hace que SCEPTRE pueda modelar numerosas no linealidades. En cada paso de integración calcula las derivadas temporales de las variables de estado independientes y obtiene después las demás variables mediante las ecuaciones de la red. Para hacerlo necesita conocer las variables independientes al comienzo del paso. Si alguna todavía no está disponible, usa su valor del paso anterior y avisa:

```
THE TERM '......' WILL CAUSE A COMPUTATIONAL DELAY
```

El error introducido es difícil de estimar: depende de la no linealidad y del tamaño del paso. No aparece con dependencias del tiempo, de tensiones de condensadores ni de corrientes de inductancias, porque estas magnitudes siempre son variables independientes. Las tensiones y corrientes de resistencias también pueden serlo si se usa la sintaxis adecuada de las secciones 2.5.3, 2.5.4 y 3.9.

Por ejemplo, puede aparecer con una fuente de corriente de diodo gobernada por tensión. Un condensador en paralelo —que además representa la capacidad real del dispositivo— proporciona una tensión de estado independiente. Si eso no basta, puede examinarse el subprograma FORTRAN `SIMUL8` generado por SCEPTRE mediante `WRITE SIMUL8 DATA` bajo `RUN CONTROLS`. Su listado muestra en qué orden se calculan las variables dependientes; quizá sea posible reformular la expresión con magnitudes que ya estén disponibles en ese punto.

Si esta definición:

```
L2, ... = X1 (P1*IR2)
```

produce el aviso `THE TERM 'IR2' WILL CAUSE A COMPUTATIONAL DELAY`, puede sustituirse por la siguiente, siempre que `VR2` ya haya sido calculada:

```
L2, ... = X1 (P1*VR2/R2)
```

3.8. Elección del método de integración

SCEPTRE ofrece tres métodos explícitos: exponencial (`XPO`), Runge–Kutta de cuarto orden (`RUK`) y trapezoidal modificado (`TRAP`); también ofrece uno implícito (`IMPLICIT`). El exponencial, desarrollado por IBM, busca un compromiso entre rapidez y precisión y se emplea automáticamente si no se elige un método. El implícito resulta ventajoso cuando la misma red contiene constantes de tiempo muy grandes y muy pequeñas.

Cada método tiene parámetros propios para limitar el tamaño del paso y estimar el error. Después de cada paso se estima el error y se calcula el tamaño siguiente. Si este cae por debajo del mínimo permitido, SCEPTRE termina con:

```
SMALLER MINIMUM STEP SIZE REQUIRED
```

El mensaje incluye el índice de la variable de estado causante; SCEPTRE lista todas las variables de estado con sus índices. Esto suele ocurrir cuando las constantes de tiempo de la red son mucho menores que el tiempo total de simulación, `STOP TIME`. Las posibilidades de corrección son:

- acortar el tiempo de simulación;
- revisar los elementos que determinan las constantes de tiempo y, si corresponde, aumentarlas;
- disminuir `MINIMUM STEP SIZE`. Probablemente haya que aumentar también el límite de pasos, `MAXIMUM INTEGRATION PASSES`.

Para detectar impulsos muy estrechos hay que limitar el tamaño del paso de forma que SCEPTRE no los salte. Antes de una simulación costosa conviene ensayar los cuatro métodos y comparar críticamente sus resultados y tiempos de cálculo: según el problema, uno puede fallar o resultar desproporcionadamente lento. El original remite a [3] y [7, cap. 3.5] para el análisis de estos métodos.

3.9. Definición de fuentes dependientes

Las fuentes dependientes deben escribirse en la forma que SCEPTRE reconoce para que estén disponibles sin retardo durante el análisis transitorio. Hay cuatro clases:

1. Fuente de tensión dependiente de una resistencia: `Exx, node1 - node2 = constant * VRyy`.
2. Fuente de corriente dependiente de una resistencia: `Jxx, node1 - node2 = constant * IRyy`.
3. Fuente de corriente de diodo de primer orden:

   ```
   Jxx, node1 - node2 = DIODE EQUATION (x1,x2)
   Jxx, node1 - node2 = DIODE TABLE xy
   ```

4. Fuente de corriente dependiente de un diodo, de segundo orden: `Jxx, node1 - node2 = variable * Jyy`, donde `Jyy` debe ser una fuente de primer orden.

La forma debe respetarse exactamente. Por ejemplo:

```
Exx, node1 - node2 = X1 (constant * VRyy)
```

puede causar un *computational delay*, porque SCEPTRE no reconoce `Exx` como dependiente de una resistencia. La definición correcta suele acortar los análisis DC y TR. En el análisis AC, una fuente dependiente mal definida se trata como fuente DC y se sustituye por cero.

A menudo puede reformularse una dependencia para ajustarla a la forma admitida. En lugar de gobernar una fuente con la tensión del condensador:

```
E1, N1 - N2 = X1 (3. * VC1)
C1, NA - NB = ....
```

se puede conectar una resistencia muy grande en paralelo y escribir:

```
E1, N1 - N2 = 3. * VR1
C1, NA - NB = ....
R1, NA - NB = grosser Wert
```

Análogamente, si una resistencia `R1` vale 2, la fuente `E1, N1 - N2 = X1 (3. * IR1)` puede escribirse como `E1, N1 - N2 = 1.5 * VR1`. La expresión alemana `grosser Wert` («valor grande») se conserva arriba tal como aparece en el ejemplo original.

3.10. Medición de corrientes y tensiones

La corriente de un componente se nombra anteponiendo `I` a su nombre y su tensión, anteponiendo `V`. Para medir sin perturbar el circuito la tensión entre nodos alejados puede conectarse una fuente de corriente nula entre ellos:

```
Jxx, NA - NZ = 0
VJxx, PLOT
```

Del mismo modo, una fuente de tensión nula permite medir la corriente de una rama sin insertar una resistencia de derivación:

```
Exx, NM - NN = 0
IExx, PLOT
```

3.11. Uso de subprogramas FORTRAN propios

Un subprograma FORTRAN permite cálculos que no se expresan directamente en SCEPTRE y puede intercambiar valores en ambas direcciones. Se llama asignando su resultado a una variable de la red:

```
networkvariable = Fname (parameterlist)
```

`Fname` puede tener como máximo seis caracteres. La función FORTRAN correspondiente comienza con `DOUBLE PRECISION FUNCTION Fname (parameterlist)`. Todas las variables de la lista de parámetros deben declararse con doble precisión, como `REAL*8` o `COMPLEX*16`. El bloque común `/CNTRLS/` permite consultar variables internas de SCEPTRE; se describe en el anexo C.1 del original. No deben modificarse dentro del subprograma las variables de SCEPTRE recibidas como argumentos.

Si se desea llamar a la función en cada paso sin intercambiar datos —por ejemplo, para escribir un archivo externo o producir un gráfico propio— puede asignarse su resultado a una variable auxiliar:

```
DEFINED PARAMETERS
  PDUMM = Fname (parameterlist)
```

Sin embargo, SCEPTRE solo genera el cálculo de `PDUMM` si la variable se utiliza en algún lugar. Para forzar la llamada en cada paso debe aparecer por segunda vez en la entrada; la forma más sencilla es incluirla bajo `OUTPUTS`:

```
OUTPUTS
  PDUMM,....
```

3.12. Notación vectorial

Mientras la red no supera los 70 elementos, los subprogramas FORTRAN generados conservan los nombres elegidos por el usuario. Por encima de ese límite, SCEPTRE pasa a una notación vectorial: almacena los elementos en `X`, los parámetros definidos en `P` e imprime una lista de correspondencias.

Este ejemplo muestra por qué una función no debe depender de un parámetro que no reciba como argumento:

```
ELEMENTS
  CX, .... = QC (VCX)
DEFINED PARAMETERS
  PL = 4711
FUNCTIONS
  QC(B) = (PL/(1.+B)*DLOG(2.+B))
```

La capacidad `CX` depende de `VCX` y `PL`, pero `PL` no se pasa a `QC`. Mientras se conservan los nombres, la variable local FORTRAN `PL` puede coincidir con el parámetro esperado. En la notación vectorial, SCEPTRE renombra el parámetro, por ejemplo como `P(2)`, mientras la función sigue usando `PL`; el valor deja de ser el pretendido. La forma segura es pasar todos los parámetros necesarios:

```
ELEMENTS
  CX, .... = QC (PL,VCX)
FUNCTIONS
  QC(A,B) = (A/(1.+B)*DLOG(2.+B))
```

3.13. Elección de las variables de estado

SCEPTRE elige las variables de estado según sus propios criterios; el orden de las instrucciones bajo `ELEMENTS` influye en esa elección. Reordenarlas y agregar `NO ELEMENT SORT` bajo `RUN CONTROLS` puede hacer que se elijan otras variables. Como el tamaño de paso también depende de esa elección y de las constantes de tiempo de la red, conviene probar estas variantes antes de una simulación extensa si se busca reducir el tiempo de cálculo.

3.14. Cálculo de integrales

Las derivadas de parámetros definidos permiten integrar una función $f(t)$ con valor inicial numérico `number`:

```
DEFINED PARAMETERS
  DPname = f(t)
  Pname = number
```

Al finalizar la simulación se obtiene $Pname = \texttt{number} + \int_0^{\texttt{STOP TIME}} f(t)\,\mathrm{d}t$. Para facilitar la evaluación cuantitativa de estas integrales, al terminar un análisis TR se imprimen los valores inicial y final de cada variable solicitada como salida. El valor inicial no puede ser otra variable.

3.15. Estadísticas de la simulación

Al terminar un análisis AC o TR, SCEPTRE imprime estadísticas y mensajes de control:

Campo	Significado
`INTEGRATION ROUTINE`	método de integración utilizado
`CURRENT SIMULATION TIME`	tiempo simulado alcanzado efectivamente
`INTEGRATION STEP COUNTER`	pasos de integración exitosos; coincide con la cantidad de resultados disponibles
`INTEGRATION PASS COUNTER`	pasos más intentos repetidos con tamaños de paso menores
`ELAPSED EXECUTION TIME [MIN]`	tiempo transcurrido de ejecución, dependiente de la carga de la computadora
`CPU EXECUTION TIME [SEC]`	tiempo efectivo de cálculo de la simulación, sin lectura de datos, compilación FORTRAN ni enlace

El campo más importante es `TERMINATION CONDITION`, que conviene comprobar primero después de cada simulación:

Mensaje	Interpretación
`STOP TIME EXCEEDED (NORMAL STOP)`	terminación normal
`MAXIMUM INTEGRATION PASSES EXCEEDED`	se necesitaron más pasos que el límite indicado
`SMALLER MINIMUM STEP SIZE REQUIRED`	se necesita un paso mínimo menor; véase la sección 3.8
`SOLUTION TIME LIMIT EXCEEDED`	se agotó el límite de tiempo de CPU
`OPTIONAL TERMINATION CONDITION MET`	se cumplió una condición de detención definida por el usuario

Para el análisis AC se imprimen además estos cuatro campos:

```
INITIAL FREQUENCY        LAST CALCULATED FREQUENCY
NUMBER FREQUENCY STEPS   TYPE FREQUENCY RUN
```

3.16. Salida impresa

Cada variable indicada bajo `OUTPUTS` puede imprimirse también como tabla. En este apartado el original afirma que el valor predeterminado de `MAXIMUM PRINT POINTS` es 0, con lo cual se suprime la impresión; `PRINT INTERVAL = number` genera la salida adicional. *Esta afirmación contradice el valor predeterminado* 1000 *indicado en el apartado 2.10.7 del mismo original*; se conservan ambos datos hasta poder verificarlos en el programa.

`PLOT INTERVAL` y `PRINT INTERVAL` pertenecen a `RUN CONTROLS`, no a `OUTPUTS`.

3.17. Condiciones iniciales

3.17.1. Punto de trabajo

Al comienzo de un análisis transitorio, las tensiones y corrientes de todos los elementos —incluidas las fuentes de corriente de diodo— valen cero si no se ha indicado otra cosa.

Cuando hay elementos que almacenan energía o fuentes de diodo, la red alcanza su punto de trabajo solo después de un tiempo de encendido suficiente. SCEPTRE puede calcularlo automáticamente mediante `RUN INITIAL CONDITIONS [ONLY]`.

Las tablas dependientes del tiempo requieren atención: para calcular el punto de trabajo se usa siempre su valor en $t = 0$. Si allí hay un salto, se toma el primer valor. Por ejemplo, la tabla `XYZ7` de la sección 2.9.2 comienza con los pares $(0, 0)$ y $(0, 1)$: SCEPTRE usa 0 como condición inicial. Si se elimina el primer par, usa 1 sin cambiar el recorrido posterior de la tensión.

También pueden darse condiciones iniciales manualmente bajo `INITIAL CONDITIONS`. Si se usan ambos métodos, esos valores sirven como semillas para el cálculo del punto de trabajo.

El cálculo automático de condiciones iniciales falla si se usan corrientes de resistencias como variables independientes dentro de tablas o ecuaciones; en ese caso puede expresarse I_x como V_x/R_x. Tampoco pueden emplearse magnitudes dependientes de capacidades o inductancias, porque estas todavía no existen durante ese cálculo. Si el problema contiene ecuaciones diferenciales expresadas mediante `P...` y `DP...`, las condiciones iniciales solo pueden determinarse por integración implícita.

3.17.2. Integración implícita

Normalmente SCEPTRE calcula iterativamente las condiciones iniciales con Newton–Raphson. Este método puede no converger o encontrar una topología inadmisible, como mallas `E-L` o nodos `J-C`. Una alternativa sería integrar transitoriamente con las excitaciones «congeladas» hasta que la red se estabilice, pero puede ser ineficiente cuando sus constantes de tiempo son muy distintas.

El método implícito de SCEPTRE realiza una especie de análisis transitorio ficticio con estos parámetros:

Parámetro	**Valor**
`MINIMUM STEP SIZE`	0
`MAXIMUM STEP SIZE`	`1E74` o `1E37`
`STARTING STEP SIZE`	`1E-8`

La integración termina cuando la derivada del vector de estado y satisface $\|\dot{y}\| \leq 10^{-8}\|y\| + 10^{-6}$. Para activar este método en lugar de Newton–Raphson se usan distintas combinaciones bajo `RUN CONTROLS`:

Solo DC, con o sin repeticiones:

```
RUN CONTROLS
  RUN INITIAL CONDITIONS ONLY
  RUN IC VIA IMPLICIT
```

DC seguido de TR:

```
RUN CONTROLS
  RUN IC VIA IMPLICIT
  STOP TIME = ...
```

DC seguido de TR, con repeticiones solo del análisis TR:

```
RUN CONTROLS
  RUN IC VIA IMPLICIT
  STOP TIME = ...
RERUN DESCRIPTION
  ...
```

DC seguido de TR, con DC y TR también en las repeticiones:

```
RUN CONTROLS
  RUN IC VIA IMPLICIT
  STOP TIME = ...
RERUN DESCRIPTION
  RUN CONTROLS
    RUN IC VIA IMPLICIT
    ...
```

3.17.3. Repeticiones

Análisis DC. Newton–Raphson usa como semillas las tensiones y corrientes declaradas bajo `INITIAL CONDITIONS`; las no indicadas comienzan en cero. Semillas adecuadas pueden acelerar mucho la convergencia. En las repeticiones pueden emplearse como semillas los resultados del primer análisis DC (`MASTER RESULTS`) o los del inmediatamente anterior (`PRECEDING RESULTS`):

```
RUN CONTROLS
  RUN INITIAL CONDITIONS ONLY
  IC FOR RERUNS = MASTER RESULTS
RERUN DESCRIPTION (n)
  ...

RUN CONTROLS
  RUN INITIAL CONDITIONS ONLY
  IC FOR RERUNS = PRECEDING RESULTS
RERUN DESCRIPTION (n)
  ...
```

En ambos ejemplos puede agregarse opcionalmente `RUN IC VIA IMPLICIT`.
Análisis TR. También pueden usarse los valores finales de un análisis transitorio como condiciones iniciales de otros análisis TR: los del primero con `IC FOR RERUNS = MASTER RESULTS`, o los del inmediatamente anterior con `IC FOR RERUNS = PRECEDING RESULTS`. El original escribe una vez `RERUN` en singular para la primera opción; aquí se usa `RERUNS`, la forma definida en los apartados anteriores.

3.18. Funciones periódicas

SCEPTRE permite generar funciones periódicas mediante expresiones aritméticas, tablas o subprogramas FORTRAN propios. También pueden tener saltos: para un mismo valor de la variable independiente se admiten dos valores dependientes, como en una función rectangular ideal.

3.18.1. Funciones ideales

Una instrucción `EXPRESSION` puede producir una función periódica ideal. Si esa función no influye directamente en la dinámica de la red —por ejemplo, si solo sirve como referencia— pueden aparecer imprecisiones en los saltos: el tamaño de paso se adapta a la dinámica de la red y el instante de integración no necesariamente coincide con el salto.
Función rectangular. Combinando `SIGN` con seno o coseno se obtienen pulsos rectangulares. La instrucción siguiente produce una tensión de 50 Hz y amplitud ±311:

```
EXXX, ....... = X1 (SIGN (311.,SIN (314.15*TIME) ) )
```

Puede añadirse un desplazamiento vertical o de fase.
Rampa periódica. El ejemplo siguiente comienza en 3,8, sube linealmente hasta 8,2 en `400E-6` unidades de tiempo y repite el ciclo:

```
ERAMP, ..... = X2 ( 3.8 + (8.2-3.8)*MOD ( (TIME/400E-6) ,1D0) )
```

Seno conmutado o señal multiplexada en el tiempo. La expresión siguiente produce el recorrido de la figura 3.1:

```
..... = X12 (SIN(6.281*TIME)*(0.5+SIGN(0.5,SIN(75.500*TIME))))
```

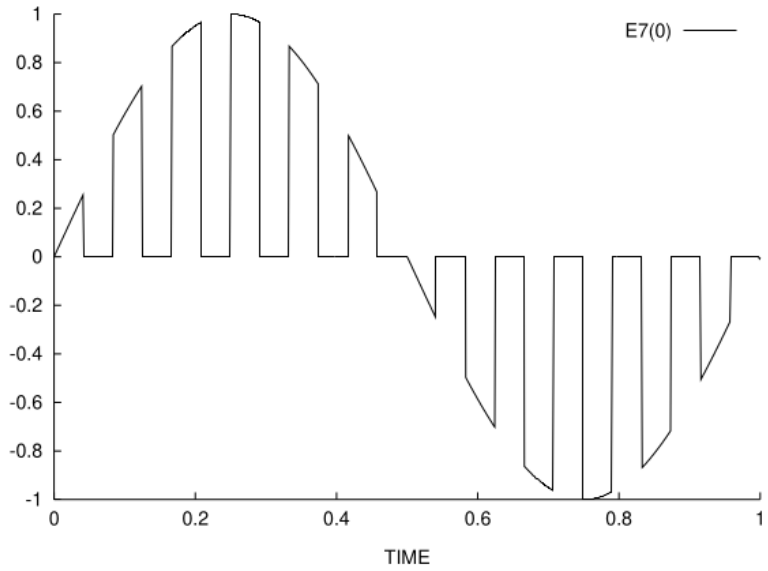


Figura 3.1: Señal sinusoidal conmutada. Gráfico reproducido del manual original.

3.18.2. `FPULSE`

Los pulsos rectangulares se usan con frecuencia para probar o accionar circuitos, pero sus saltos empeoran mucho la convergencia. Una alternativa es un pulso trapezoidal con tiempos de subida y bajada finitos. `FPULSE` genera el pulso periódico de la figura 3.2 y se invoca como una función FORTRAN:

```
.... = FPULSE (v1, v2, td, tr, tf, pw, per)
```

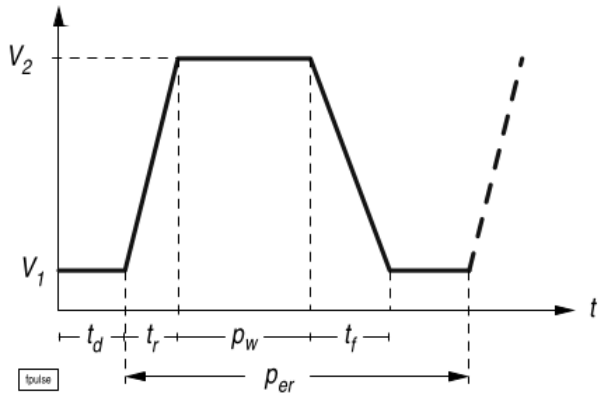


Figura 3.2: Parámetros del pulso periódico `FPULSE`. Diagrama reproducido del manual original.

Los parámetros se identifican en el diagrama; `td` es un retardo inicial que se aplica una sola vez.

3.18.3. `FSINSQ`

`FSINSQ` reemplaza los flancos de un pulso rectangular ideal por tramos de $\sin^2$, como muestra la figura 3.3. Se llama con los mismos argumentos de `FPULSE`:

```
.... = FSINSQ (v1, v2, td, tr, tf, pw, per)
```

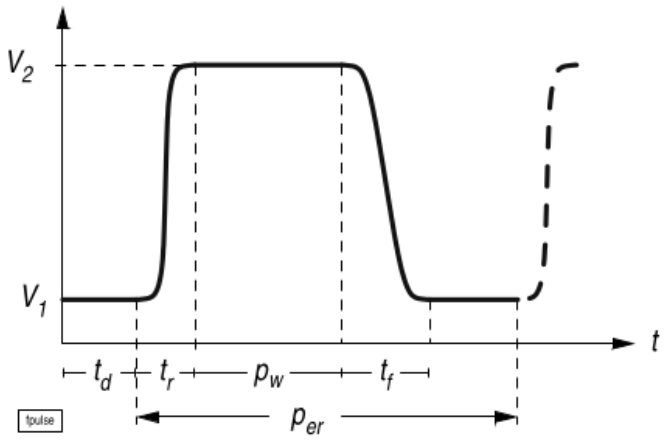


Figura 3.3: Parámetros del pulso periódico `FSINSQ`. Diagrama reproducido del manual original.

También aquí `td` es un retardo inicial único.

3.19. Nombres de variables reservados

Un nombre elegido por el usuario puede entrar en conflicto con una variable interna de SCEPTRE; a veces el problema solo aparece al compilar el FORTRAN generado. El

original recomienda evitar `PS` junto con `DPS`.

SCEPTRE distingue mayúsculas de minúsculas en nombres libres, pero algunos compiladores FORTRAN convierten todas las minúsculas a mayúsculas. Por eso `PMI` y `Pmi` pueden terminar siendo el mismo nombre y causar errores. Asimismo, ciertos nombres de parámetros auxiliares de `EQUATION` pueden confundirse con variables internas; en lugar de `Q9 (A,B,C,...)` es preferible `Q9 (AA,BB,CC,...)`.

3.20. Fuentes de corriente alterna

Las fuentes de corriente alterna deben llevar a la derecha del signo igual dos datos separados por coma y encerrados entre paréntesis; su formato difiere del de una fuente transitoria. Si se omiten los paréntesis, SCEPTRE ignora la fuente sin emitir un error. Ejemplo literal del original:

```
E1, Gnd - 2.2 = (325.27,0.)
```

4 Ejemplos

Los ejemplos como parte viva del libro

Nota editorial de esta traducción. Los listados completos de este capítulo no están duplicados en el texto: cada entrada SCEPTRE es un archivo `.d` en `docs/manual-es/ejemplos/`. La instrucción LaTeX `\lstinputlisting` incorpora su contenido al compilar el libro. Por tanto, el programa que se lee en estas páginas es el mismo archivo que puede entregarse a SCEPTRE; una corrección del `.d` se refleja en la próxima edición del PDF. Se conservaron en el texto solo dos extractos de salida y un modelo aislado, que no constituyen por sí mismos entradas ejecutables.

Archivo	**Función**
`ejemplos/aNN.d`	Entrada ejecutable de SCEPTRE, incorporada al libro mediante `\lstinputlisting`.
`ejemplos/aNN.gp`	Instrucciones de Gnuplot para los casos que producen series.
`graficos/*.pdf`	Figuras calculadas desde esos casos e incluidas en estas páginas.
`build/`	Archivos auxiliares de compilación; no son parte de la fuente del libro.

El recorrido reproducible es `.d` ⟶ SCEPTRE ⟶ resultados ⟶ Gnuplot ⟶ figuras ⟶ PDF. Desde la raíz de `sceptre2novender`, los siguientes comandos regeneran las siete figuras calculadas y luego el libro:

```
cd docs/manual-es
make graficos
make
```

Para ejecutar un caso por separado conviene copiarlo a un directorio de trabajo: el lanzador `sc1` escribe allí archivos intermedios además de los resultados. Este ejemplo ejecuta A03 sin modificar `ejemplos/a03.d`; supone que ya se está en `docs/manual-es` y conserva los resultados en el directorio temporal que imprime al final:

```
lanzador="$(pwd)/../../nuevo/bin/sc1"
trabajo="$(mktemp -d)"
cp ejemplos/a03.d "$trabajo/a03.d"
(cd "$trabajo" && SCEPTRE_NO_CONF=1 "$lanzador" a03.d)
echo "Resultados: $trabajo"
```

La opción `SCEPTRE_NO_CONF=1` evita que una configuración personal cambie la prueba. En A03, A07, A08, A09 y A10, la instrucción `RUN INITIAL CONDITIONS ONLY` pide puntos de trabajo o análisis sobre ellos, no una evolución en tiempo ni un barrido de frecuencia. Sus resultados escalares aparecen en el listado `aNN.log`; que no se produzca una serie `.simul` ni una figura no indica un fallo de SCEPTRE. A01, A02, A04, A05, A11 y A13 sí producen series y sus figuras se regeneran con `make graficos`. Así, los ejemplos y sus gráficos pueden volver a comprobarse cuando cambie SCEPTRE, sin convertir las imágenes históricas en una fuente de datos.

4.1. A01: inversor con carga RC

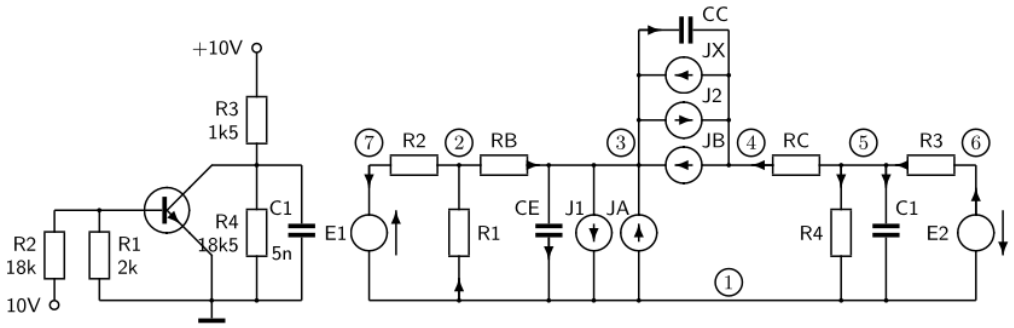


Figura 4.1: Circuito inversor y circuito equivalente preparado para SCEPTRE.

La figura 4.1 muestra un inversor sencillo cargado por una red RC, junto con el circuito equivalente preparado para SCEPTRE. El objetivo es estudiar el efecto de la radiación gamma sobre su funcionamiento.

El transporte de carga inducido por la radiación en el semiconductor se representa mediante la fuente de corriente `JX`, dependiente del tiempo. Su evolución se define en `TABLE 1`. Las condiciones iniciales —el punto de funcionamiento— se fijan explícitamente; solo se efectúa un análisis transitorio.

Para el transistor se usa el circuito equivalente de Ebers–Moll. Las dos fuentes controladas por tensión `J1` y `J2` describen los diodos mediante la función integrada `DIODE EQUATION (x1,x2)`. La corriente del diodo se calcula a partir de su tensión según

$$J = x_1(\exp(x_2 V_J) - 1).$$

Para comparar distintos perfiles de `JX`, el caso incluye dos repeticiones con valores diferentes de `TABLE 1`. Las figuras 4.2 y 4.3 muestran parte de los resultados. La entrada SCEPTRE se conserva en `ejemplos/a01.d` y se incorpora a continuación sin modificar sus datos ni sus comentarios:

```
CIRCUIT DESCRIPTION
A01: INVERTER CIRCUIT LOADED WITH RC NETWORK
  ELEMENTS
    E1, 7-1 = 10
    E2, 1-6 = 10
    CE, 3-1 = EQUATION 1 (5., 70., J1)
    CC, 3-4 = EQUATION 1 (8., 370., J2)
    C1, 5-1 = 500
    R1, 1-2 = 2
    R2, 2-7 = 17
    R3, 6-5 = 1.5
    R4, 5-1 = 18.5
    RB, 2-3 = .3
    RC, 5-4 = .015
    J1, 3-1 = DIODE EQUATION (1.E-7, 35.)
    J2, 3-4 = DIODE EQUATION (5.E-7, 37.)
    JA, 1-3 = .1 * J2
    JB, 4-3 = .98 * J1
    JX, 4-3 = TABLE 1 (TIME)
  OUTPUTS
    VCE, VCC, VC1, IR3, J1
  INITIAL CONDITIONS
```

```
    VC1 = 9.25
    VCE = -1
    VCC = -10.25
  FUNCTIONS
    TABLE 1
      0,     0
      40,   .8
      100, .5
      200, .25
      500,   0
      600,   0
    EQUATION 1 (AA,BB,CC) = (AA + BB * CC)
  RUN CONTROLS
    INTEGRATION ROUTINE = TRAP       ; original intgr. routine
    ; INTEGRATION ROUTINE = IMPLICIT   ; much more faster
    STOP TIME = 800
RERUN DESCRIPTION (2)
  FUNCTIONS
    TABLE 1 = 0,      0,     0
              40,    1.2,    .4
             100,    .75,   .25
             200,   .375, .125
             500,      0,     0
             600,      0,     0
END
```

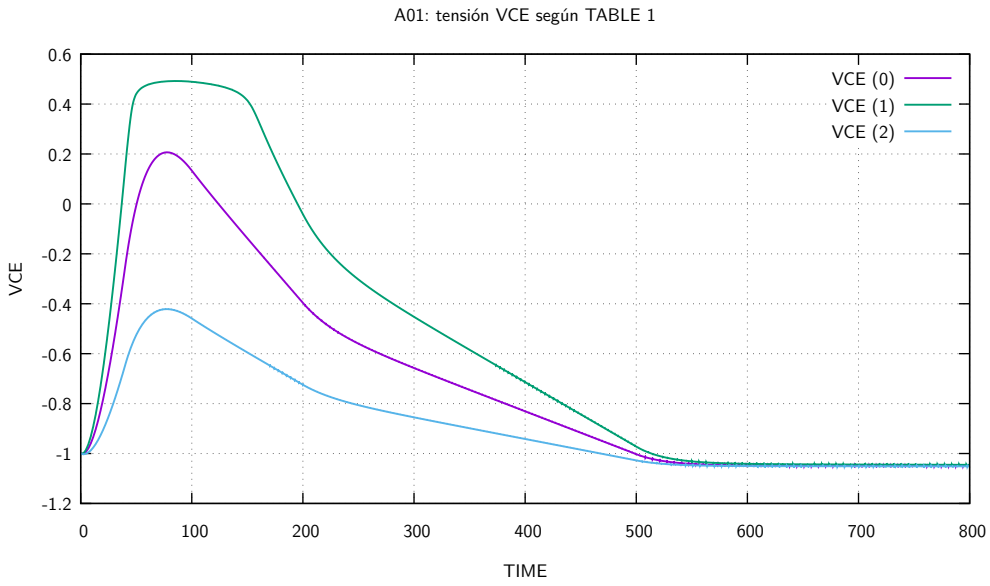


Figura 4.2: Tensión `VCE` con los distintos perfiles de `JX` definidos por `TABLE 1`; gráfico regenerado con SCEPTRE a partir de `a01.d`.

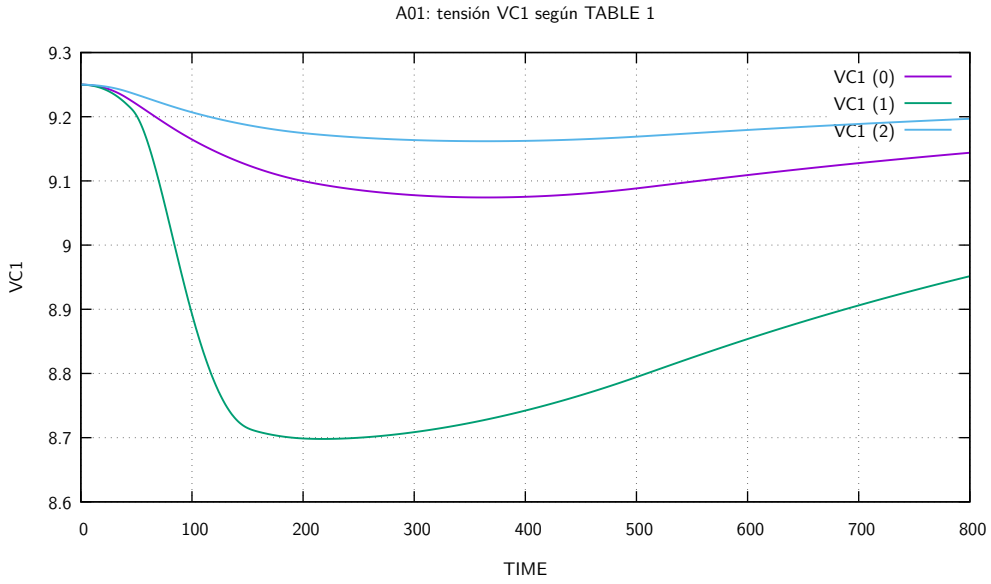


Figura 4.3: Tensión `VC1` con los distintos perfiles de `JX` definidos por `TABLE 1`; gráfico regenerado con SCEPTRE a partir de `a01.d`.

4.2. A02: seguidor de emisor con transformador

La figura 4.4 muestra una combinación de un seguidor de emisor y una etapa de emisor común que alimenta un transformador con carga resistiva. La entrada es una tensión en rampa, `E1`. La figura incluye el modelo simplificado de transistor utilizado para ambas etapas. Los diodos `J1` y `J2` se describen mediante `DIODE TABLE 1` y `DIODE TABLE 2`. El transistor `T2` tiene una ganancia de corriente menor que `T1`: 0,975 frente a 0,98. Las inductancias `L1` y `L2` se acoplan mediante la inductancia mutua `M`.

Se estudia el circuito ya en funcionamiento, sin simular el transitorio producido al conectar la alimentación. Por eso se calcula primero el punto de funcionamiento con `RUN INITIAL CONDITIONS`; ese resultado se usa como estado inicial del análisis transitorio, cuyo fin determina `STOP TIME`. El original informa:

```
I/C transient values at time equals zero:
VCET1 = 7.3886840E-01
VCCT1 =-1.2777978E+01
VCET2 = 7.3864447E-01
VCCT2 =-8.9324536E+00
IL1   = 3.1299703E+00
IL2   =-0.0000000E+00
```

La simulación posterior produce los perfiles de tensión `VL1` y `VL2` de la figura 4.5.

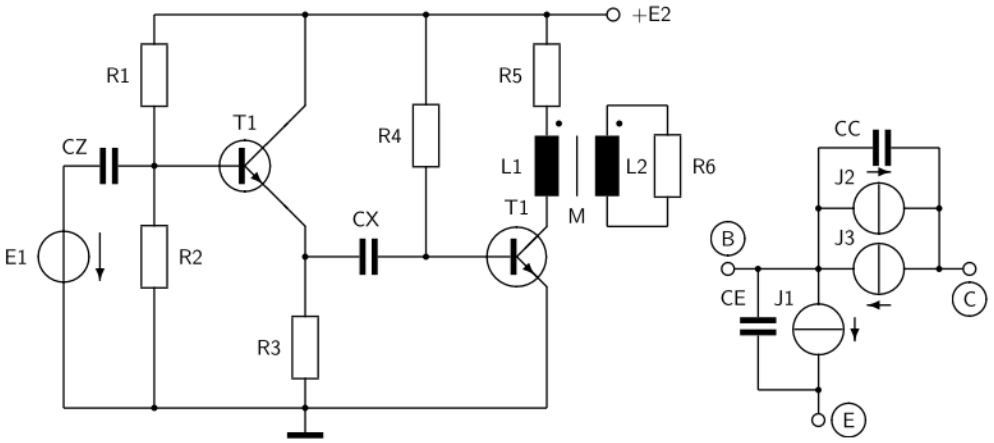


Figura 4.4: Seguidor de emisor con transformador y circuito equivalente del transistor.

El modelo `2N9999AA` se define una vez en `MODEL DESCRIPTION` y se utiliza dos veces en `CIRCUIT DESCRIPTION`. Incluye dos capacidades dependientes de tensión, `CC` y `CE`, gobernadas por `EQUATION 1`; sus argumentos son dos constantes y el valor de una de las tablas de diodo. Estas tablas también determinan las fuentes de corriente primarias `J1` y `J2`. La fuente secundaria `J3` vale `P1*J1`. Al instanciar `T2`, el parámetro `P1` cambia de 0,98 a 0,975.

Nota de traducción: el texto introductorio del original dice que el modelo se guarda «permanentemente», pero el listado lo declara `MODEL 2N9999AA (TEMP)`. Se conserva la declaración del listado sin resolver aquí esa contradicción.

```
MODEL DESCRIPTION
MODEL 2N9999AA (TEMP) (B-E-C)
ELEMENTS
```

```
  CE,B-E=EQUATION 1(5.,40.,TABLE 1(VCE))
  CC,B-C=EQUATION 1(10.,400.,TABLE 2(VCC))
  J1,B-E=DIODE TABLE 1
  J2,B-C=DIODE TABLE 2
  J3,C-B=P1*J1
DEFINED PARAMETERS
  P1=.98
FUNCTIONS
  EQUATION 1(AA,BB,CC)=(AA+BB*CC)
  DIODE TABLE 1
    0,0,.3,0,.65,.05,.7,.6,.72,1.4,.73,2,.74,3.4,.77,10,.8,22
  DIODE TABLE 2
    0,0,.58,0,.62,.4,.64,1,.66,2,.67,3,.69,7,.7,12
CIRCUIT DESCRIPTION
A02: transformer coupled amplifier
ELEMENTS
  E1,1-2=TABLE 1(TIME)
  DERIVATIVE E1=TABLE DE1
  E2,1-4=20
  CZ,2-3=1E3
  CX,5-6=1E3
  R1,4-3=30
  R2,3-1=20
  R3,5-1=2
  R4,4-6=240
  R5,4-7=3.3
  R6,9-1=1.8
  T1,3-5-4=MODEL 2N9999AA (TEMP)
  T2,6-1-8=MODEL 2N9999AA (TEMP,CHANGE P1=.975)
  L1,7-8=100
  L2,9-1=900
  M,L1-L2=299.7
OUTPUTS
  VR6,VL1,VL2
FUNCTIONS
  TABLE 1
    0,0,50,.5,100,.5
  TABLE DE1
    0,.01,50,.01,50,0,100,0
RUN CONTROLS
  STOP TIME=500
  RUN INITIAL CONDITIONS
END
```

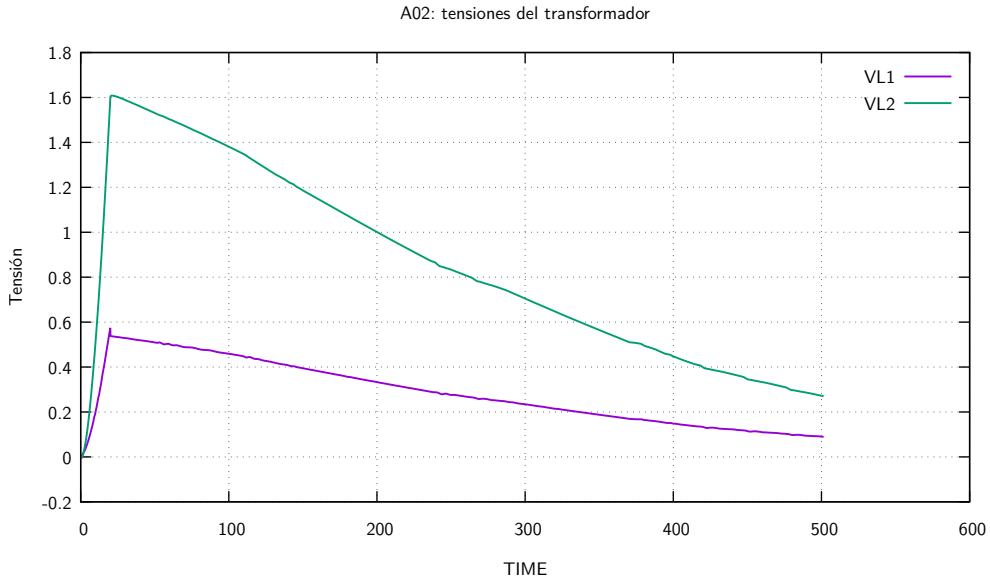


Figura 4.5: Tensiones `VL1` y `VL2` en el transformador; gráfico regenerado desde `a02.d`.

En las salidas se solicitan las tensiones de los devanados primario y secundario, `VL1` y `VL2`; esta última coincide con `VR6`. También se solicita `VR6`. El párrafo inglés del original afirma que se usa la integración implícita, pero el listado no incluye `INTEGRATION ROUTINE = IMPLICIT`. Según la sección 3.7, sin selección explícita se usa `XPO`; por tanto, no puede deducirse del listado que este resultado se haya obtenido con el método implícito.

4.3. A03: circuito Darlington (análisis DC)

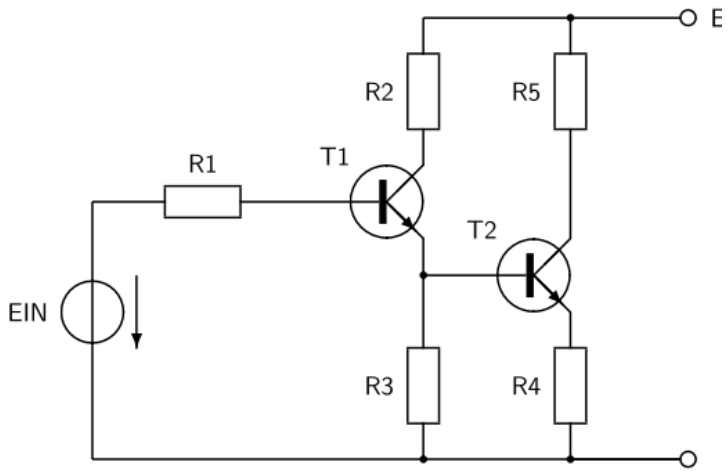


Figura 4.6: Circuito Darlington.

El circuito de la figura 4.6 se analiza para distintas ganancias de corriente del transistor `T1`. Se buscan las tensiones en `R4` y `R5`, así como la potencia absorbida. Basta un análisis de corriente continua mediante `RUN INITIAL CONDITIONS ONLY`. El modelo de transistor `2N706A` ya existe en la biblioteca de modelos, de modo que no forma parte de la entrada del circuito; el original lo reproduce para facilitar la lectura:

```
MODEL 2N706A (PERM) (B-E-C)
  ELEMENTS
    CE, 1-E = Q1(5.,70.,J1)
    CC, 1-2 = Q1(8.,370.,J2)
    RB, B-1 = .3
    RC, C-2 = .015
    J1, 1-E = DIODE EQUATION(1.E-7,35.)
    J2, 1-2 = DIODE EQUATION(5.E-7,37.)
    JA, E-1 = .1*J2
    JB, 2-1 = P1*J1
    JX, 2-1 = 0
  DEFINED PARAMETERS
    P1 = 0.98
  OUTPUTS
    VCE,VCC,J1,PLOT
  FUNCTIONS
    Q1(A,B,C) = (A+B*C)
```

La ganancia de corriente se representa mediante `P1`. Para modificar solo la instancia `T1`, el nombre interno del parámetro se acompaña del nombre de esa instancia: `P1T1`. El caso principal y sus tres repeticiones son:

```
CIRCUIT DESCRIPTION
Example 3: Darlington pair
  ELEMENTS
    EC , 1-6    = 10
```

```
    EIN , 1-2   = 1
    R1 , 2-3    = 20
    R2 , 6-5    = 5
    R3 , 4-1    = 200
    R4 , 8-1    = .4
    R5 , 6-7    = 1
    T1 , 3-4-5 = MODEL 2N706A (PERM)
    T2 , 4-8-7 = MODEL 2N706A (PERM)
  DEFINED PARAMETERS
    PEC = X1 (EC*IEC)
    PEIN= X2 (EIN*IEIN)
  OUTPUTS
    PEC,PEIN,VR3,VR4
  RUN CONTROLS
    RUN INITIAL CONDITIONS ONLY
RERUN DESCRIPTION (3)
  DEFINED PARAMETERS
    P1T1= .96, .93, .9
END
```

La salida reúne variables pedidas tanto en el circuito principal como en el modelo. Las condiciones iniciales publicadas son:

Ejecución	VCET1	VCCT1	VCET2	VCCT2
Inicial	3.3818865E-01	-8.9376695E+00	4.4221916E-01	-8.8226781E+00
Repetición 1	3.3769191E-01	-8.9456210E+00	4.4163421E-01	-8.8381601E+00
Repetición 2	3.3696571E-01	-8.9570427E+00	4.4077602E-01	-8.8603292E+00
Repetición 3	3.3626101E-01	-8.9679012E+00	4.3993970E-01	-8.8813264E+00

Los resultados del cálculo del punto de funcionamiento, en el mismo orden, son:

Ejecución	PEC	PEIN	VR3	VR4
Inicial	5.3007344E+00	2.7598821E-04	6.5620879E-01	2.1082735E-01
Repetición 1	5.1910074E+00	5.4289718E-04	6.5128728E-01	2.0655488E-01
Repetición 2	5.0340188E+00	9.2654405E-04	6.4422545E-01	2.0044292E-01
Repetición 3	4.8854795E+00	1.2915639E-03	6.3752024E-01	1.9466077E-01

4.4. A04: circuito equivalente de pequeña señal

La figura 4.7 representa un amplificador lineal de dos etapas acopladas mediante redes RC. Se estudia su respuesta a una entrada sinusoidal `E1` de 1 mV y 100 kHz. Para cada transistor se utiliza un modelo temporal de parámetros h. Como solo interesa la respuesta de corriente alterna, la fuente de tensión continua se cortocircuita. La simulación termina a los 30 000 ns o antes si la tensión de salida en `RL2` alcanza 20 V.

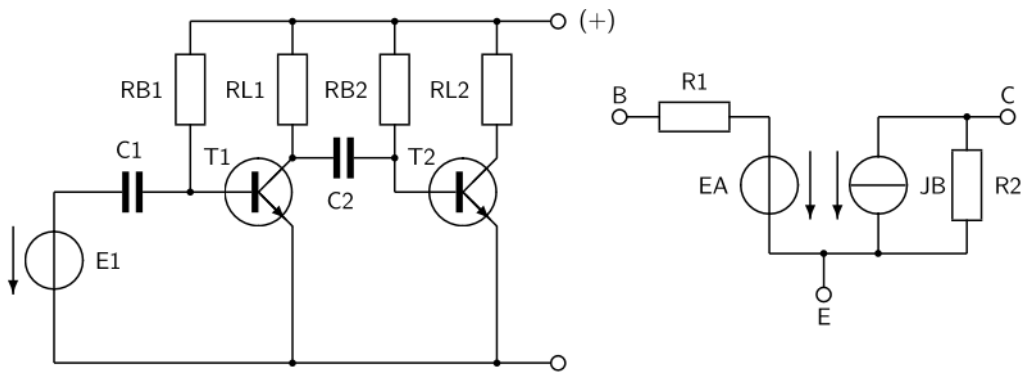


Figura 4.7: Amplificador de pequeña señal y circuito equivalente de parámetros h.

```
MODEL DESCRIPTION
  MODEL SS1 (TEMP) (B-E-C)
    ELEMENTS
      EA, E-X = .0005 * VR2
      R2, C-E = 2000
      R1, B-X = .3
      JB, C-E = 50. * IR1
CIRCUIT DESCRIPTION
A04 : small signal equivalent circuit
  ELEMENTS
    E1, 1-2 = X1(.001*DSIN(.000628*TIME))
    C1, 2-3 = 5E6
    C2, 4-5 = 5E6
    RB1,3-1 = 100
    RL1,1-4 = 1
    RB2,5-1 = 100
    RL2,1-6 = 1
    T1, 3-1-4 = MODEL SS1
    T2, 5-1-6 = MODEL SS1
  OUTPUTS
    VRL1, VRL2, VC1, VC2, PLOT
  RUN CONTROLS
    STOP TIME = 30000
    INTEGRATION ROUTINE = TRAP
    TERMINATE IF (VRL2 .GE. 20.)
END
```

La figura 4.8 presenta las tensiones en `RL1` y `RL2`. Se conserva la doble escala vertical de la gráfica original.

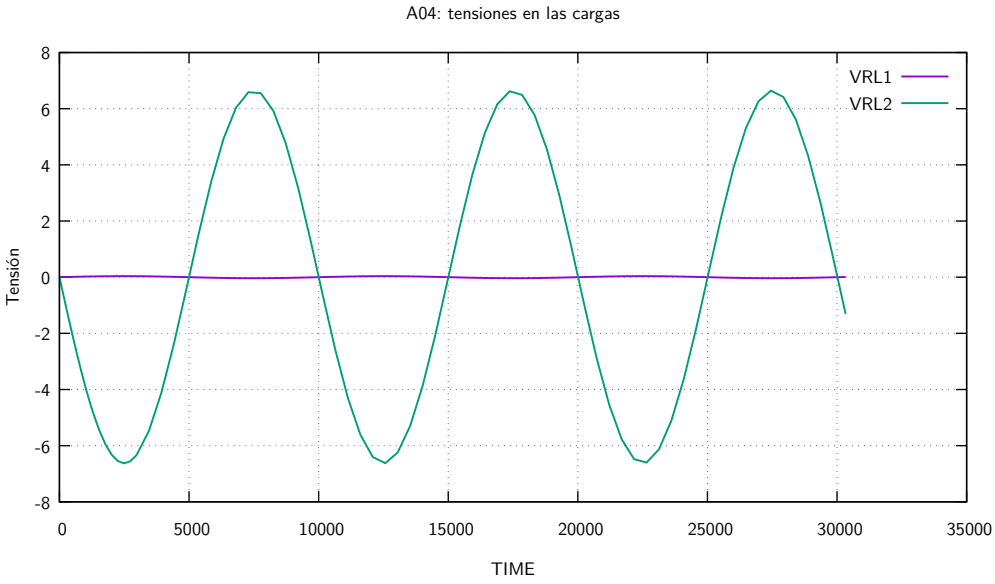


Figura 4.8: Evolución de las tensiones `VRL1` y `VRL2`; gráfico regenerado desde `a04.d`.

4.5. A05: sistema de ecuaciones diferenciales

Este ejemplo resuelve un sistema de tres ecuaciones diferenciales de primer orden utilizando parámetros definidos. El sistema no depende de ninguna red eléctrica:

$$\begin{aligned}\dot{x} &= -6x + 5y + 10,\\ \dot{y} &= 5x - 7y + 2z,\\ \dot{z} &= 0{,}2y - 0{,}2z - 0{,}5.\end{aligned}$$

Las condiciones iniciales $x(0) = 6$, $y(0) = 5$ y $z(0) = 4$ se introducen mediante `PX`, `PY` y `PZ`. Después de cada paso de integración se les asignan valores nuevos.

```
CIRCUIT DESCRIPTION
A05: Solution of simultaneous differential equations
DEFINED PARAMETERS
  DPX = EQUATION 1 (PX, PY)
  DPY = EQUATION 2 (PX, PY, PZ)
  DPZ = EQUATION 3 (PY, PZ)
  PX = 6
  PY = 5
  PZ = 4
OUTPUTS
  PX(X), PY(Y), PZ(Z), XSTPSZ, PLOT
FUNCTIONS
  EQUATION 1 (AA,BB) = (-6.*AA+5.*BB+10.)
  EQUATION 2 (AA,BB,CC) = (5.*AA-7.*BB+2.*CC)
  EQUATION 3 (AA,BB) = (.2*AA-.2*BB-.5)
RUN CONTROLS
  INTEGRATION ROUTINE = TRAP
  STOP TIME = 100
END
```

La figura 4.9 muestra las soluciones $x(t)$, $y(t)$ y $z(t)$. En la salida para impresora, `PX`, `PY` y `PZ` se renombran `X`, `Y` y `Z`; según la nota del original, este cambio de nombre no afecta la salida gráfica producida mediante NGP. También se solicita `XSTPSZ`, una variable interna de SCEPTRE que contiene el tamaño del paso de integración; véase [8, anexo B.2] del original.

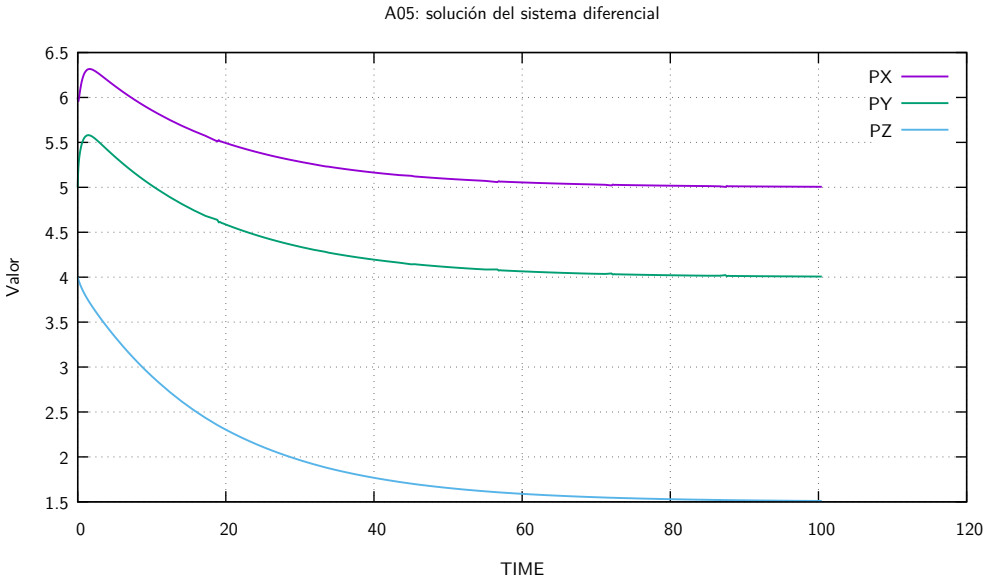


Figura 4.9: Evolución de `PX`, `PY` y `PZ`; gráfico regenerado desde `a05.d`.

4.6. Análisis de tolerancias de un circuito Darlington

Los ejemplos siguientes aplican distintos análisis de tolerancias al circuito Darlington de A03 (sección 4.3). Aunque SCEPTRE permite efectuarlos en una sola simulación, aquí se presentan por separado.

4.6.1. A07: análisis Monte Carlo

La variable independiente es el factor de ganancia de corriente `P1` del modelo `2N706A`. Su valor nominal es 0,96 y se admite una variación entre 0,9 y 0,98. En `MONTE CARLO`, antes de la barra se enumeran las variables que se observarán; después aparecen los parámetros que se variarán dentro de esos límites, `P1T1` y `P1T2`. Se utiliza la distribución gaussiana predeterminada.

```
MODEL DESCRIPTION
MODEL 2N706A (TEMP) (B-E-C)
  ELEMENTS
    CE, 1-E = Q1(5.,70.,J1)
    CC, 1-2 = Q1(8.,370.,J2)
    RB, B-1 = .3
    RC, C-2 = .015
    J1, 1-E = DIODE EQUATION(1.E-7,35.)
    J2, 1-2 = DIODE EQUATION(5.E-7,37.)
    JA, E-1 = .1*J2
    JB, 2-1 = P1*J1
    JX, 2-1 = 0
  DEFINED PARAMETERS
    P1 = 0.96 (0.98, 0.9)
  FUNCTIONS
    Q1(A,B,C) = (A+B*C)
  OUTPUTS
    VCE, VCC, J1
CIRCUIT DESCRIPTION
A07: Darlington pair, MONTE CARLO
  ELEMENTS
    EC , 1-6     = 10
    EIN , 1-2    = 1
    R1 , 2-3     = 20
    R2 , 6-5     = 5
    R3 , 4-1     = 200
    R4 , 8-1     = .4
    R5 , 6-7     = 1
    T1 , 3-4-5 = MODEL 2N706A
    T2 , 4-8-7 = MODEL 2N706A
  MONTE CARLO
    (VCET1,VCCT1,J1T1,VCET2,VCCT2,
     J1T2,PEC,PEIN,VR3,VR4/P1T1,P1T2)
  DEFINED PARAMETERS
    PEC = X1 (EC*IEC)
    PEIN= X2 (EIN*IEIN)
  OUTPUTS
    PEC,PEIN,VR3,VR4
  RUN CONTROLS
    RUN MONTE CARLO = 10
    RUN INITIAL CONDITIONS ONLY
END
```

Al añadir a un nombre el identificador de instancia del modelo se puede acceder a sus elementos y variables dentro del circuito completo: por ejemplo, `P1T1` designa `P1` en `T1`. El

cálculo inicial publicado da `VCET1 = 3.5123970E-01`, `VCCT1 = -8.9126370E+00`, `VCET2 = 4.3874672E-01` y `VCCT2 = -8.9197511E+00`.

Para ambas variables independientes, SCEPTRE informa un valor nominal de 0,96, límite inferior de 0,9, límite superior de 0,98, media teórica de 0,94 y desviación estándar teórica de 0,013333333. El número aleatorio inicial de la ejecución publicada es `127263527`; después de las diez iteraciones, el número final es `1007121511`.

Las estadísticas observadas de las variables independientes son:

Variable	**Nominal**	**Mínimo**	**Máximo**	**Media muestral**
`P1T1`	0,96000000	0,93082453	0,97906312	0,95388296
`P1T2`	0,96000000	0,91926269	0,95900296	0,94255822

Las desviaciones estándar muestrales respectivas son 0,016480857 y 0,011320930. Para las variables dependientes, el original informa:

Variable	**Nominal**	**Mínimo**	**Máximo**	**Media**	**Desv. est.**
`VCET1`	3.5123970E-01	3.5109792E-01	3.6510355E-01	3.5782344E-01	4.0157705E-03
`VCCT1`	-8.9126370E+00	-8.9210128E+00	-8.8596399E+00	-8.8927857E+00	2.2013686E-02
`J1T1`	2.1824751E-02	2.1716713E-02	3.5455596E-02	2.7726820E-02	3.9307541E-03
`VCET2`	4.3874672E-01	4.3440367E-01	4.3795162E-01	4.3649493E-01	1.4850834E-03
`VCCT2`	-8.9197511E+00	-9.0300613E+00	-8.9392260E+00	-8.9773819E+00	3.5350405E-02
`J1T2`	4.6675032E-01	4.0093037E-01	4.5394055E-01	4.3189852E-01	2.2252503E-02
`PEC`	4.6903307E+00	4.0210736E+00	4.5616323E+00	4.3372422E+00	2.2675435E-01
`PEIN`	8.7254006E-04	6.3257750E-04	1.8368401E-03	1.2575918E-03	4.3076374E-04
`VR3`	6.3104773E-01	6.0275509E-01	6.2660595E-01	6.1664745E-01	9.7793613E-03
`VR4`	1.8670015E-01	1.6037217E-01	1.8157624E-01	1.7275943E-01	8.9010012E-03

La salida presenta asimismo un último conjunto de resultados del cálculo de condiciones iniciales:

Variable	**Valor**
`VCET1`	3.6510355E-01
`VCCT1`	-8.8596399E+00
`J1T1`	3.5455596E-02
`VCET2`	4.3445810E-01
`VCCT2`	-9.0300613E+00
`J1T2`	4.0169492E-01
`PEC`	4.0323994E+00
`PEIN`	1.4793579E-03
`VR3`	6.0486548E-01
`VR4`	1.6067799E-01

4.6.2. A08: análisis de sensibilidad

El análisis de sensibilidad estima, mediante derivadas parciales, la influencia de elementos y parámetros independientes sobre las variables dependientes del circuito. Para el Darlington se piden dos conjuntos de derivadas bajo `SENSITIVITY`; no es necesario declarar tolerancias. SCEPTRE calcula directamente las derivadas de las variables propias de la red, pero para cada parámetro definido dependiente `Pnombre` el usuario debe proporcionar su diferencial total `GPnombre`.

```
MODEL DESCRIPTION
MODEL 2N706A (TEMP) (B-E-C)
  ELEMENTS
    CE, 1-E = Q1(5.,70.,J1)
    CC, 1-2 = Q1(8.,370.,J2)
    RB, B-1 = .3
    RC, C-2 = .015
    J1, 1-E = DIODE EQUATION(1.E-7,35.)
    J2, 1-2 = DIODE EQUATION(5.E-7,37.)
    JA, E-1 = .1*J2
    JB, 2-1 = P1*J1
    JX, 2-1 = 0
  DEFINED PARAMETERS
    P1 = 0.96
  FUNCTIONS
    Q1(A,B,C) = (A+B*C)
  OUTPUTS
    VCE, VCC, J1
CIRCUIT DESCRIPTION
Example A08: Darlington pair, SENSITIVITY
  ELEMENTS
    EC , 1-6    = 10
    EIN , 1-2   = 1
    R1 , 2-3    = 20
    R2 , 6-5    = 5
    R3 , 4-1    = 200
    R4 , 8-1    = .4
    R5 , 6-7    = 1
    T1 , 3-4-5 = MODEL 2N706A
    T2 , 4-8-7 = MODEL 2N706A
  SENSITIVITY
    (VCET1,VCCT1,J1T1,PR3,PRR3/P1T1,P1T2,R3)
    (VCET2,VCCT2,PEIN,VR4/P1T1,P1T2,R3,R4)
  DEFINED PARAMETERS
    PEIN= X2 (EIN*IEIN)
    GPEIN=P4*DIEIN
    P4 = X4 (EIN)
    PR3 = XR3 (R3*IR3*IR3)
    GPR3 = PDR3*DR3+PDIR3*DIR3
    PDR3 = XDR3 (IR3*IR3)
    PDIR3= XDIR3 (2.*IR3*R3)
    PRR3 = XRR3 (IR3*VR3)
    GPRR3= PDIRR3*DIR3+PDVRR3*DVR3
    PDIRR3 = XDIRR3 (VR3)
    PDVRR3 = XDVRR3 (IR3)
  OUTPUTS
    VR3,VR4
  RUN CONTROLS
    RUN SENSITIVITY
    RUN INITIAL CONDITIONS ONLY
END
```

Se estudia, entre otras magnitudes, la sensibilidad de la potencia de entrada `PEIN` y de la potencia disipada en `R3`. Para esta última se usan dos expresiones equivalentes, una basada en la corriente y otra en la caída de tensión. A partir de $P_{\mathrm{EIN}} = I_{\mathrm{EIN}} E_{\mathrm{IN}}$ se obtiene

$$\mathrm{d}P_{\mathrm{EIN}} = \frac{\partial P_{\mathrm{EIN}}}{\partial I_{\mathrm{EIN}}}\,\mathrm{d}I_{\mathrm{EIN}} + \frac{\partial P_{\mathrm{EIN}}}{\partial E_{\mathrm{IN}}}\,\mathrm{d}E_{\mathrm{IN}}.$$

Como E_{IN} es constante y no figura entre las variables independientes, el último término desaparece; la entrada SCEPTRE utiliza `P4 = X4 (EIN)` y `GPEIN = P4*DIEIN`.

Para la potencia expresada mediante la corriente,

$$P_{\mathrm{R3}} = I_{\mathrm{R3}}^2 R_3, \qquad \mathrm{d}P_{\mathrm{R3}} = 2 I_{\mathrm{R3}} R_3\,\mathrm{d}I_{\mathrm{R3}} + I_{\mathrm{R3}}^2\,\mathrm{d}R_3.$$

Los factores de ambos términos se definen como `PDIR3` y `PDR3`, y el diferencial como `GPR3 = PDR3*DR3 + PDIR3*DIR3`. Si la potencia se expresa a partir de la tensión,

$$P_{\mathrm{RR3}} = I_{\mathrm{R3}} V_{\mathrm{R3}}, \qquad \mathrm{d}P_{\mathrm{RR3}} = V_{\mathrm{R3}}\,\mathrm{d}I_{\mathrm{R3}} + I_{\mathrm{R3}}\,\mathrm{d}V_{\mathrm{R3}}.$$

En la entrada, `PDIRR3` y `PDVRR3` contienen los factores correspondientes, y `GPRR3` reúne el diferencial completo.

La salida publicada presenta para cada pareja de variables una derivada parcial y su sensibilidad normalizada. El primer conjunto usa `P1T1`, `P1T2` y `R3` como variables independientes:

Dependiente	Independiente	Derivada parcial	Sensibilidad normalizada
VCET1	P1T1	2.23659125E-01	6.11299797E-01
	P1T2	-4.07175175E-02	-1.11288150E-01
	R3	-3.75988248E-06	-2.14092110E-03
VCCT1	P1T1	1.23628942E+00	-1.33163490E-01
	P1T2	-1.24486175E-01	1.34086834E-02
	R3	-1.14951356E-05	2.57951394E-04
J1T1	P1T1	1.70846451E-01	7.51498104E+00
	P1T2	-3.11028820E-02	-1.36811486E+00
	R3	-2.87206067E-06	-2.63192979E-02
PR3	P1T1	2.07019909E-01	9.98134165E+01
	P1T2	5.27852161E-01	2.54500777E+02
	R3	5.68080410E-06	5.70617891E-01
PRR3	P1T1	1.03764443E-01	5.00294086E+01
	P1T2	2.64134241E-01	1.27350752E+02
	R3	-2.11814175E-06	-2.12760299E-01

El segundo conjunto añade `R4` a esas variables independientes:

Dependiente	Independiente	Derivada parcial	Sensibilidad normalizada
VCET2	P1T1	1.04334237E-02	2.28288587E-02
	P1T2	2.66473139E-02	5.83056702E-02
	R3	7.88042611E-07	3.59224392E-04
	R4	-1.11083071E-02	-1.01273074E-02
VCCT2	P1T1	2.44690518E-01	-2.63351403E-02
	P1T2	1.09869936E+00	-1.18248970E-01
	R3	1.84816183E-05	-4.14397623E-04
	R4	2.06232105E-01	-9.24833451E-03
PEIN	P1T1	-1.49908934E-02	-1.64935209E+01
	P1T2	-1.24411528E-03	-1.36882044E+00
	R3	-1.14882427E-07	-2.63328717E-02
	R4	-7.18255438E-03	-3.29271044E+00
VR4	P1T1	6.81772684E-02	3.50563074E-01
	P1T2	1.74127029E-01	8.95349841E-01
	R3	5.14946906E-06	5.51629887E-03
	R4	3.94163071E-01	8.44483683E-01

El cálculo de condiciones iniciales que acompaña esta salida informa:

Variable	Valor
VCET1	3.5123970E-01
VCCT1	-8.9126370E+00
J1T1	2.1824751E-02
VCET2	4.3874672E-01
VCCT2	-8.9197511E+00
J1T2	4.6675032E-01
VR3	6.3104773E-01
VR4	1.8670015E-01

4.6.3. A09: análisis de peor caso

El análisis de peor caso busca los extremos de una función objetivo —una variable dependiente de la red— al variar uno o más parámetros independientes dentro de límites de tolerancia declarados. En el ejemplo, `P1` de cada transistor puede variar de 0,9 a 0,98; `R3`, de 195 a 205; y `R4`, de 0,35 a 0,45.

```
MODEL DESCRIPTION
MODEL 2N706A (TEMP) (B-E-C)
  ELEMENTS
    CE, 1-E = Q1(5.,70.,J1)
    CC, 1-2 = Q1(8.,370.,J2)
    RB, B-1 = .3
    RC, C-2 = .015
    J1, 1-E = DIODE EQUATION(1.E-7,35.)
    J2, 1-2 = DIODE EQUATION(5.E-7,37.)
    JA, E-1 = .1*J2
    JB, 2-1 = P1*J1
  DEFINED PARAMETERS
    P1 = 0.96 (0.98, 0.9)
  FUNCTIONS
    Q1(A,B,C) = (A+B*C)
  OUTPUTS
    VCE, VCC, J1
CIRCUIT DESCRIPTION
Example A09: Darlington pair, WORST CASE
  ELEMENTS
    EC , 1-6     = 10
    EIN , 1-2    = 1
    R1 , 2-3     = 20
    R2 , 6-5     = 5
    R3 , 4-1     = 200 (195,205)
    R4 , 8-1     = .4 (0.35,0.45)
    R5 , 6-7     = 1
    T1 , 3-4-5 = MODEL 2N706A
    T2 , 4-8-7 = MODEL 2N706A
  WORST CASE
    (VCET1,VCCT1,J1T1,PEIN/P1T1,P1T2,R3)
    (VCET2,VR4/P1T2,R3,R4)
  DEFINED PARAMETERS
    PEIN= X2 (EIN*IEIN)
    GPEIN=P4*DIEIN
    P4 = X4 (EIN)
  OUTPUTS
    VR3,VR4
  RUN CONTROLS
    RUN WORST CASE
    RUN INITIAL CONDITIONS ONLY
END
```

La barra en cada conjunto `WORST CASE` separa las funciones objetivo de los parámetros que pueden variar. Para la función objetivo `PEIN`, que es un parámetro definido y no una variable propia de la red, también se proporciona su diferencial total `GPEIN`, como en A08.

El original muestra parte de la búsqueda de los extremos de `VCCT1` al variar `P1T1`, `P1T2` y `R3`. Con los valores nominales se obtiene `VCCT1 = -8.91263697E+00`. Los valores y componentes del gradiente inicial son:

Variable	Nominal	Gradiente	Límite inf.	Límite sup.
P1T1	9.60000000E-01	1.23628942E+00	9.00000000E-01	9.80000000E-01
P1T2	9.60000000E-01	-1.24486175E-01	9.00000000E-01	9.80000000E-01
R3	2.00000000E+02	-1.14951356E-05	1.95000000E+02	2.05000000E+02

La salida señala que, siguiendo el gradiente desde el punto nominal, se llega al límite inferior de `P1T1` a una distancia `-6.030340E-02`, y al límite superior a una distancia `2.010113E-02`. Los extremos calculados para `VCCT1` son:

	Nominal	Máximo	Mínimo
VCCT1	-8.91263697E+00	-8.89478115E+00	-8.95520062E+00
P1T1	9.60000000E-01	9.80000000E-01	9.00000000E-01
P1T2	9.60000000E-01	9.57986132E-01	9.66041604E-01
R3	2.00000000E+02	2.00000000E+02	2.00000001E+02

En el máximo, los componentes del gradiente para `P1T1`, `P1T2` y `R3` son, respectivamente, `1.60698069E+00`, `-1.76735527E-01` y `-1.61045665E-05`; en el mínimo son `6.78924171E-01`, `-4.67066273E-02` y `-4.44565760E-06`.

4.6.4. A10: optimización

El ejemplo busca minimizar las funciones objetivo `VR3` y `PEC` respecto de `P1T1`, `P1T2` y `EC`. Los parámetros independientes deben tener límites de tolerancia: los dos primeros pueden variar entre 0,9 y 0,98, mientras que `EC` puede variar entre 9,5 y 10,5.

```
MODEL DESCRIPTION
MODEL 2N706A (TEMP) (B-E-C)
  ELEMENTS
    CE, 1-E = Q1(5.,70.,J1)
    CC, 1-2 = Q1(8.,370.,J2)
    RB, B-1 = .3
    RC, C-2 = .015
    J1, 1-E = DIODE EQUATION(1.E-7,35.)
    J2, 1-2 = DIODE EQUATION(5.E-7,37.)
    JA, E-1 = .1*J2
    JB, 2-1 = P1*J1
  DEFINED PARAMETERS
    P1 = 0.96 (0.98, 0.9)
  FUNCTIONS
    Q1(A,B,C) = (A+B*C)
  OUTPUTS
    VCE, VCC, J1
CIRCUIT DESCRIPTION
Example A10: Darlington pair, OPTIMIZATION
  ELEMENTS
    EC , 1-6     = 10 (9.5, 10.5)
    EIN , 1-2    = 1
    R1 , 2-3     = 20
    R2 , 6-5     = 5
    R3 , 4-1     = 200
    R4 , 8-1     = .4
    R5 , 6-7     = 1
    T1 , 3-4-5 = MODEL 2N706A
    T2 , 4-8-7 = MODEL 2N706A
  OPTIMIZATION
    (VR3,PEC/P1T1,P1T2,EC)
  DEFINED PARAMETERS
    PEC = XEC (EC*IEC)
    GPEC= P3*DIEC+P5*DEC
    P3 = X3 (EC)
    P5 = X5 (IEC)
  OUTPUTS
    VR3,VR4
  RUN CONTROLS
    RUN OPTIMIZATION = 40
    RUN INITIAL CONDITIONS ONLY
END
```

Aquí los parámetros definidos intervienen tanto como variable dependiente (`PEC`) cuanto como independientes (`P1` en cada transistor). Para `PEC` debe darse el diferencial completo: a partir de $P_{\mathrm{EC}} = E_{\mathrm{C}} I_{\mathrm{EC}}$, `P3` representa el factor E_{C} de `DIEC` y `P5` el factor I_{EC} de `DEC`. El original advierte que un parámetro definido solo puede actuar como variable independiente si describe una fuente de primer o segundo orden. Para la interpretación detallada de los parámetros de optimización, remite a [2, 3].

La salida reproducida en el original corresponde a la función objetivo `VR3`. Declara tres variables independientes, cero pasos aleatorios, un factor inicial de la matriz H igual a 1, criterio de convergencia 10^{-7}, control de tamaño del paso aleatorio 0,2, estimación mínima de la función igual a 0 y determinante inicial de H igual a 1. La aproximación inicial de H

es la matriz identidad de orden tres.

Variable	Nominal	Límite inferior	Límite superior
P1T1	9.60000000E-01	9.00000000E-01	9.80000000E-01
P1T2	9.60000000E-01	9.00000000E-01	9.80000000E-01
EC	1.00000000E+01	9.50000000E+00	1.05000000E+01

La optimización termina normalmente después de 13 pasadas de las 40 permitidas; el valor final de VR3 es 5.65021325E-01. Los parámetros finales y sus gradientes son:

Variable	Valor	Gradiente	Valor transformado (rad)	Gradiente transformado
P1T1	9.00000002E-01	4.50846113E-01	-2.86201981E-04	-5.16132196E-06
P1T2	9.00000004E-01	5.48451029E-01	-4.44554674E-04	-9.75265843E-06
EC	1.00000000E+01	1.28031073-147	1.57079633E+00	6.40155365-148

Nota de traducción: los dos últimos números de la fila EC aparecen en el original como 1.28031073-147 y 6.40155365-148, sin la letra E antes del exponente. Se reproducen literalmente.

4.6.5. A11: circuito equivalente de pequeña señal (AC)

La figura 4.10 muestra un circuito con transistor cuyo comportamiento en frecuencia se estudia alrededor de su punto de operación.

4.7. A11: Kleinsignal-Ersatzschaltung (AC)

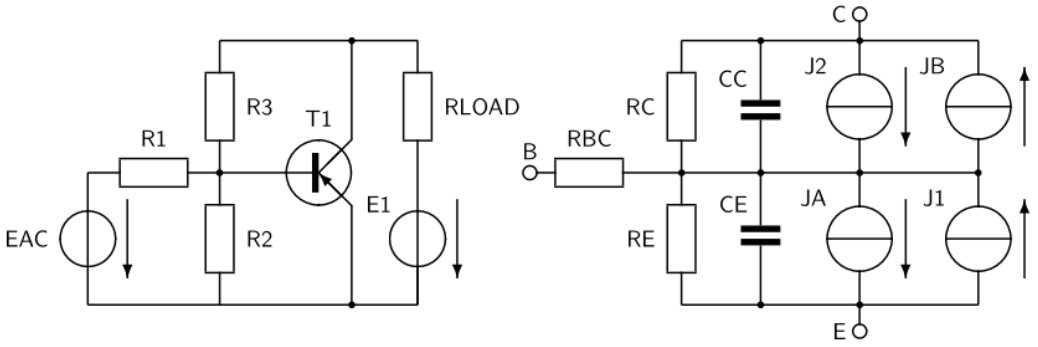


Figura 4.10: Circuito de pequeña señal y circuito equivalente del transistor (figura 4.10 del original).

```
MODEL DESCRIPTION
  MODEL 101M (B-C-E)
    ELEMENTS
      RBC, BR-B = 50.
      RC,   C -BR = 2.4E9
      RE,   E -BR = 5.7E8
      CC,   C -BR = Q1 (5.5E-12, .8, VCC, .3, 2.88E-7, J2, 3.12E-12)
      CE,   E -BR = Q2 (3.5E-12, 17E-9, J1)
      JA,    BR-E = .332 * J2
      JB,    BR-C = .986 * J1
      J1,    E -BR = DIODE EQUATION (60E-9, 38.4)
      J2,    C -BR = DIODE EQUATION (54.3E-8, 38.4)
    FUNCTIONS
      Q1 (AA,AB,AC,AD,AE,AF,AG) = (AA/(AB-AC)**AD + AE*(AF+AG))
      Q2 (AA,AB,AC)              = (AA+AB*AC)
    OUTPUTS
      VCC, VCE, NYQUIST
      CC, CE
      JA, JB, DEGREES
      J1, J2, RADIANS
CIRCUIT DESCRIPTION
A11: small signal AC-Analysis
around a circuit's DC operating point
  ELEMENTS
    EAC, 1 - B = (0.2, 0.)
    R1,    B - 2 = 25E4
    R2,    2 - 1 = 2000
    R3,    2 - 3 = 11500
    RLOAD,3 - 4 = 100
    E1,    1 - 4 = -20
    T1,    2-3-1 = MODEL 101M
  OUTPUTS
    VR2, VR3, COMPLEX
  RUN CONTROLS
    RUN INITIAL CONDITIONS
    RUN AC
```

```
    PRINT EIGENVALUES
    PRINT EIGENVECTORS
    INITIAL FREQUENCY = 1E5
    FINAL FREQUENCY = 1E6
    NUMBER FREQUENCY STEPS = 20
    TYPE FREQUENCY RUN = LINEAR
END
```

Se combinan los análisis de corriente continua (DC) y alterna (AC). En el primero se ponen a cero todas las fuentes de AC; en el segundo, todas las fuentes de DC. Por eso conviene desacoplar las fuentes de AC y DC mediante un capacitor o una resistencia grande, como `R1` en este ejemplo. Durante el análisis de AC, las fuentes de diodo se sustituyen por resistencias correspondientes al punto de operación del transistor.

La parte de los resultados reproducida en el original comienza con los valores iniciales y las salidas antes del barrido de frecuencia:

```
I/C transient values at time equals zero:
VCCT1 =-1.3023927E+01
VCET1 = 3.6223429E-01

IC outputs before AC run
VCCT1        -1.3023927E+01
VCET1         3.6223429E-01
CCT1          2.3449600E-12
CET1          1.1243506E-09
JAT1         -1.8027600E-07
JBT1          6.5009336E-02
J1T1          6.5932389E-02
J2T1         -5.4300000E-07
VR2          -4.0836859E-01
VR3           1.2977793E+01

J9 source J1T1 is replaced by resistor = 0.3949750
J9 source J2T1 is replaced by resistor = 0.2400000E+13
```

Los valores propios complejos calculados son puramente reales; las partes imaginarias impresas valen cero:

Número	Parte real	Parte imaginaria	Módulo
1	-0.65292E+10	0.00000E+00	0.65292E+10
2	-0.33353E+08	0.00000E+00	0.33353E+08

La traza y la suma de los valores propios son, respectivamente:

Traza	`-6.5625163884676018E+09`
Suma de valores propios	`-6.5625163884676008E+09`

La parte imaginaria de la suma es nula. El mayor error al comparar AS con $S\Lambda$ es `(5.69969E-07, 0)`; al multiplicar la matriz modal por su inversa, los mayores errores son `(0.11102E-15, 0)` en la diagonal y `(0.17639E-13, 0)` fuera de ella.

	Columna 1	Columna 2
Matriz modal S (partes reales)		
Fila 1	-1.0000E+00	-9.9999E-01
Fila 2	2.0843E-03	-3.9932E-03
Matriz modal inversa S^{-1} (partes reales)		
Fila 1	-6.5705E-01	1.6454E+02
Fila 2	-3.4296E-01	-1.6454E+02

En todas estas entradas de ambas matrices, la parte imaginaria impresa es cero. La figura 4.11 representa las tensiones complejas `VCCT1` y `VCET1` como un diagrama de Nyquist. Hay que distinguir las asignaciones de sus respectivos ejes de parte real (X) e imaginaria (Y), así como sus diferentes escalas.

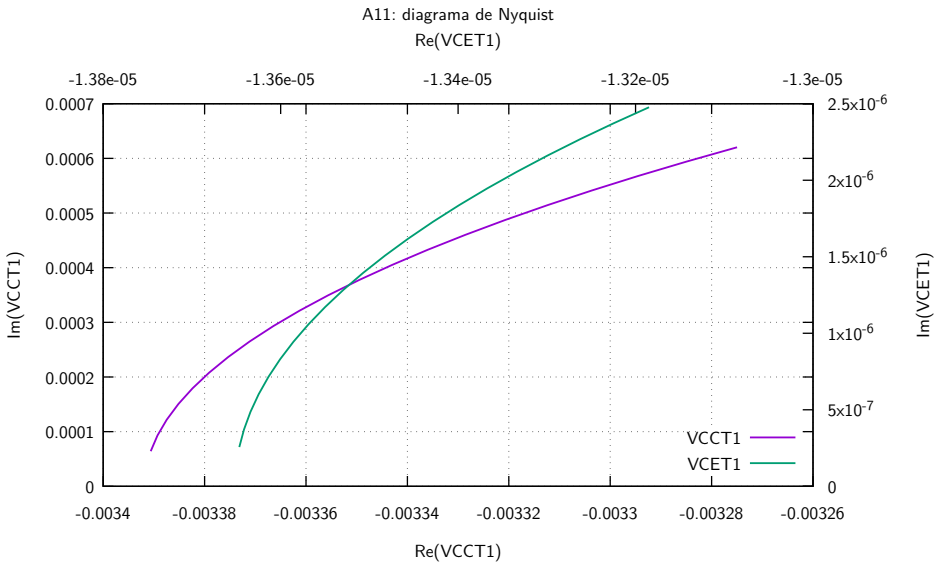


Figura 4.11: Diagrama de Nyquist de `VCCT1` y `VCET1`; gráfico regenerado desde `a11.d` con las dos escalas del original.

4.6.6. A13: actuador electrohidráulico

El circuito de la figura 4.12 representa un actuador electrohidráulico [5, cap. 9.7]. Los autores de ese capítulo son A. Brent White y Michael A. Chipman, del AFWL en Albuquerque, Nuevo México.

A13: Elektro-hydraulischer Stellantrieb

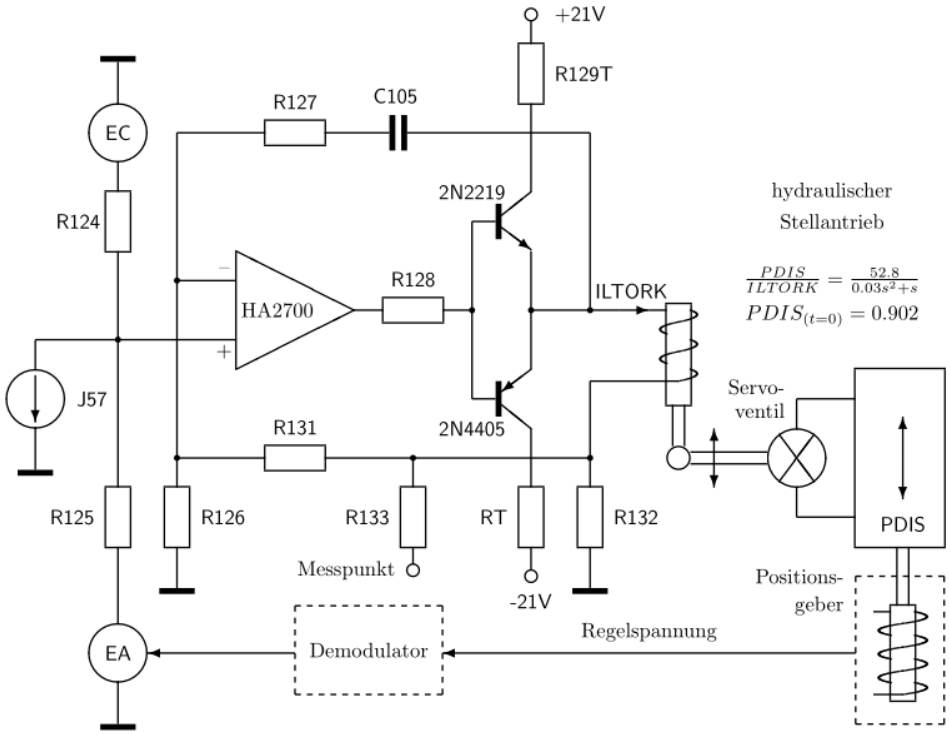


Figura 4.12: Circuito del lazo de control (figura 4.12 del original).

La consigna se establece mediante la fuente de tensión `EC = TABLE C (TIME)`. El amplificador operacional y los dos transistores amplifican la diferencia entre consigna y señal de retorno; mediante una armadura magnética, esta diferencia acciona una servoválvula. Un captador inductivo convierte el error de regulación en una tensión, que se demodula y se realimenta al regulador con signo contrario por medio de la fuente controlada `EA`.

`PDIS` es la variable regulada y su evolución se representa en la figura 4.13. Una función de transferencia relaciona la corriente de la bobina `ILTORK` con `PDIS`; SCEPTRE permite calcularla directamente mediante los parámetros definidos `PX1`, `PX2` y sus derivadas `DPX1`, `DPX2`. Las fuentes de corriente nula permiten medir tensiones entre nodos arbitrarios sin pérdidas ni efectos sobre el circuito: `J57`, por ejemplo, proporciona la señal diferencial `VJ57` a la entrada del amplificador operacional.

```
MODEL DESCRIPTION
MODEL HA 2700 (N-I-OUT-GND)
ELEMENTS
  EA1,    GND-3 = PAO*VRIN
  EA3,    GND-12 = TABLE 1 (VCA2)
  CIN,      N-I = 1.E-12
  RIN,      N-I = 1.E6
  CA1,      4-6 = 2280.E-6
  CA2,      5-GND= 45.E-12
```

```
  RA1,     3-4 = 100.
  RA2,     4-5 = 100.E3
  RA6,    12-OUT= 100.
  RA7,     6-GND= .028
DEFINED PARAMETERS
  PAO = 2.E5
FUNCTIONS
TABLE 1
  -20,-13, -13,-13, 13,13, 20,13

MODEL 2N2219 (B-E-C)
ELEMENTS
  CC, B1-C1 = 13.E-12
  CE, B1-E = 22.E-12
  RB, B-B1 = 50.
  RC, C-C1 = 2.8
  JE, B1-E = DIODE EQUATION (3.02E-14,40.)
  JC, B1-C1 = DIODE EQUATION (1.19E-13,38.)
  JN, C1-B1 = PAO*JE
  JI, E-B1 = PAI*JC
DEFINED PARAMETERS
  PAO =.9927
  PAI = XAI (0.702*PAO)

MODEL 2N4405 (B-E-C)
ELEMENTS
  CE, E-B1 = 55E-12
  CC, C1-B1 = 43.E-12
  RB, B1-B = 20.
  RC, C1-C = 1.4
  JE, E-B1 = DIODE EQUATION (1.02E-13,39.)
  JC, C1-B1 = DIODE EQUATION (1.71E-13,38.)
  JN, B1-C1 = PAO*JE
  JI, B1-E = PAI*JC
DEFINED PARAMETERS
  PAO = .9914
  PAI = XAI (0.6829*PAO)

CIRCUIT DESCRIPTION
A13: torquer closed-loop step response test circuit
     parameters for a 2% step increase from 88% point in displacement schedule
     Jensen, McNamee: Handbook of Circuit Analysis Languages and Techniques
     simplified, without radiation effects, etc.
ELEMENTS
  J57, 57-0 = 0
  JT,    73-0 = 0
  JDR, 72-0 = 0
  JR1, 73-82 = 0
  JR2, 83-0 = 0.
  JOT, 71-0 = 0
  O3,57-70-71-0 = MODEL HA 2700
  EC,     0-56 = TABLE C (TIME)
  R124, 56-57 = 23.7E3
  R125, 57-11 = 20000
  EA,     0-11 = XA (PDIS*1.843)
  R126, 70-0 = 10E3
  C105, 73-81 = 4700E-12
  R127, 81-70 = 800E3
  R128, 71-72 = 511
  R129T,80-79 = 95.
  R131, 70-82 = 66.5E3
```

```
  R132, 82-0 = 10.
  R133, 83-82 = 10E3
  RTORK,73-TI = 50
  LTORK,82-TI = .2
  CTORK,73-82 = .127E-6
  T4,72-73-74 = MODEL 2N4405
  T3,72-73-79 = MODEL 2N2219
  E13,    0-80 = 21
  E14, 78-0 = 21.
  RT,    74-78 = 60.
DEFINED PARAMETERS
  PX1 = .902
  PX2 = 0.
  DPX1 = PX2
  DPX2 = XXX ((-1./.03)*PX2+1760.*ILTORK)
  PDIS = PX1
  PSTP = XSTP (LOG10(XSTPSZ))
INITIAL CONDITIONS
  ILTORK = 0
  VCCT3 = -2.1E1
  VCCT4 = -2.09162E1
  VCTORK = 0
OUTPUTS
  VJ57,EA,EC,VJOT,VJDR,VJT,ILTORK,PDIS,VJR2,VJR1,PSTP
FUNCTIONS
  TABLE C
  0,-1.97, .01,-1.97, .01,-2.07, 1.,-2.07
RUN CONTROLS
  INTEGRATION ROUTINE = IMPLICIT
  STOP TIME = .45
  MINIMUM STEPSIZE = 1E-20
END
```

Este ejemplo interdisciplinario ilustra cómo SCEPTRE combina magnitudes eléctricas y mecánicas con funciones de transferencia. La referencia [5, cap. 9.7] presenta una explicación más detallada.

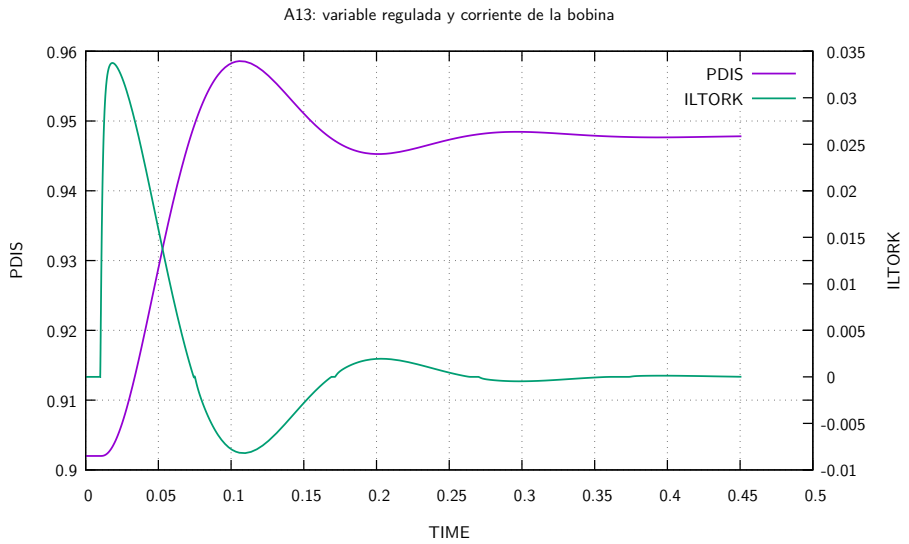


Figura 4.13: Variable regulada `PDIS` y corriente de la bobina `ILTORK`; gráfico regenerado desde `a13.d`.

Parte I

Descripción matemática (Volumen II)

5 La teoría de SCEPTRE

Esta parte del manual describe los algoritmos matemáticos que utiliza SCEPTRE. Los capítulos siguientes son, en esencia, una traducción del Volumen II del manual original [3]. No se tratan los procedimientos específicos de cada computadora.

5.1. Panorama general

Un programa que estudia la evolución temporal de un sistema dinámico se encarga de formular matemáticamente el sistema y de plantear sus ecuaciones diferenciales.

5.2. Planteamiento general de la solución

SCEPTRE ofrece tres métodos de análisis de una red:

1. **Análisis de corriente continua (DC).** Determina las tensiones de la red que servirán de condiciones iniciales para un análisis transitorio o de corriente alterna posterior. El tiempo se considera constante, no una variable independiente. Los resultados pueden utilizarse por sí mismos o pasar automáticamente como valores iniciales al análisis posterior; en este último caso proporcionan las condiciones iniciales del sistema de ecuaciones diferenciales que se resolverá.

2. **Análisis transitorio (TR).** Toma el tiempo como variable independiente y resuelve el sistema de ecuaciones diferenciales en función de él para obtener la evolución temporal de la red. Puede seguir al análisis de corriente continua o emplearse solo, si ya se conocen las condiciones iniciales.

3. **Análisis de corriente alterna (AC).** Estudia la respuesta en frecuencia de la red: toma la frecuencia como variable independiente y resuelve el sistema de ecuaciones correspondiente. También puede acompañar al análisis de corriente continua o realizarse de forma independiente.

El procedimiento general que sigue utiliza notación común a los tres análisis. En todos ellos se debe construir un *árbol de la red*, aunque las reglas para hacerlo varían según el método. El árbol conecta todos los nodos sin formar mallas cerradas. Sus conexiones se llaman *ramas del árbol*; las demás son *ramas de enlace*. Un *árbol C* contiene elementos preferentemente en el orden E, C, R y L; no contiene fuentes de corriente, que deben quedar en las ramas de enlace. Un *conjunto de corte* está formado por los elementos cuya eliminación divide la red en dos grupos de nodos.

Con el árbol se construye la matriz B. Esta permite expresar las tensiones de las ramas de enlace mediante tensiones de las ramas del árbol, y las corrientes de las ramas del árbol mediante corrientes de las ramas de enlace. La figura 5.1 del original muestra la matriz compuesta de todas las clases de elementos:

$$B = \begin{array}{c|ccccc} & 4 & 5 & 6 & 7 & Y \\ \hline 1 & B_{14} & B_{15} & B_{16} & B_{17} & B_{1Y} \\ 2 & B_{24} & B_{25} & B_{26} & B_{27} & B_{2Y} \\ 3 & B_{34} & B_{35} & B_{36} & B_{37} & B_{3Y} \\ 8 & B_{84} & B_{85} & B_{86} & B_{87} & B_{8Y} \\ 9 & B_{94} & B_{95} & B_{96} & B_{97} & B_{9Y} \\ 0 & B_{04} & B_{05} & B_{06} & B_{07} & B_{0Y} \\ X & B_{X4} & B_{X5} & B_{X6} & B_{X7} & B_{XY} \end{array}$$

La tabla 5.1 clasifica todos los elementos de red de SCEPTRE. Las clases 1, 2, 3, 8, 9, 0 y X aparecen como filas de la matriz compuesta; las clases 4, 5, 6, 7 e Y, como columnas.

Clase	**Elementos**
1	Capacitores en ramas de enlace
2	Resistencias en ramas de enlace
3	Inductancias en ramas de enlace
4	Capacitores en ramas del árbol
5	Resistencias en ramas del árbol
6	Inductancias en ramas del árbol
7	Fuentes de tensión
8	Fuentes de corriente independientes
9	Fuentes de corriente de primer orden, dependientes de tensión[a]
0	Fuentes de corriente de segundo orden, dependientes de otras fuentes de corriente[b]
Y	Fuentes de tensión controladas por tensiones en resistencias[c]
X	Fuentes de corriente controladas por corrientes en resistencias[d]
D	Fuentes de tensión de redes adjuntas con fuentes de corriente de segundo orden

Tabla 5.1: Clasificación de los elementos de red (tabla 5.1 del original).

[a] Solo en el cálculo de condiciones iniciales y el análisis AC. [b] Solo en el cálculo de condiciones iniciales. [c] Necesarias para los análisis temporal y de frecuencia. [d] Necesarias para los análisis TR y AC.

De esta clasificación se deducen las matrices siguientes:

R_{22}	Diagonal de las resistencias de las ramas de enlace.
$G_{22} = R_{22}^{-1}$	Inversa de esa diagonal.
R_{55}	Diagonal de las resistencias de las ramas del árbol.
$G_{55} = R_{55}^{-1}$	Inversa de esa diagonal.
C_{11}	Diagonal de los capacitores de las ramas de enlace.
$S_{11} = C_{11}^{-1}$	Inversa de esa diagonal.
C_{44}	Diagonal de los capacitores de las ramas del árbol.
$S_{44} = C_{44}^{-1}$	Inversa de esa diagonal.
L_{33}	Inductancias y acoplamientos mutuos dentro y entre ramas de enlace.
L_{36}	Acoplamientos mutuos entre ramas del árbol y de enlace.
L_{66}	Inductancias del árbol y acoplamientos mutuos entre sus ramas.
G_{99}	Diagonal de las derivadas de las tensiones de fuentes de corriente de primer orden.

Para cada clase $k \in \{1, 2, 3, 4, 5, 6, 8, 9, 0\}$, los vectores I_k y V_k contienen las corrientes y tensiones de los elementos correspondientes. Así, las clases 1 y 4 son capacitores de enlace y de árbol; 2 y 5, resistencias; 3 y 6, inductancias; 8, fuentes de corriente independientes; 9 y 0, fuentes de corriente de primer y segundo orden. E_7 es el vector de las fuentes de tensión.

5.3. Solución transitoria

5.3.1. Operaciones matriciales

Un sistema que evoluciona en el tiempo puede describirse mediante variables de estado. Si se conocen las tensiones V_4 de los capacitores del árbol y las corrientes I_3 de las inductancias de enlace, pueden determinarse todas las demás tensiones y corrientes de la red. Esas dos clases de variables permiten formular un sistema de ecuaciones diferenciales de primer orden. A partir de V_4 e I_3 se calculan V_1, V_2, V_3, V_5, V_6, I_1, I_2, I_4, I_5, I_6, I_7 y V_8. La integración numérica también requiere las derivadas de las variables de estado para obtener los valores del paso siguiente.

Parte de las ecuaciones necesarias se deduce directamente de la matriz B del análisis transitorio. Se construye a partir del árbol C y, a diferencia de la matriz compuesta, tiene algunas submatrices nulas y no distingue fuentes de distintas clases:

$$B_{\text{TR}} = \begin{array}{c|cccc} & 4 & 5 & 6 & 7 \\ \hline 1 & B_{14} & 0 & 0 & B_{17} \\ 2 & B_{24} & B_{25} & 0 & B_{27} \\ 3 & B_{34} & B_{35} & B_{36} & B_{37} \\ 8 & B_{84} & B_{85} & B_{86} & B_{87} \end{array}$$

Por ejemplo, $B_{15} = 0$: las propiedades del árbol C impiden que un capacitor de enlace forme una malla con una resistencia del árbol. Las tensiones de las ramas de enlace quedan expresadas mediante las tensiones del árbol:

$$V_1 = -B_{14}V_4 - B_{17}E_7, \tag{5.1}$$
$$V_2 = -B_{24}V_4 - B_{25}V_5 - B_{27}E_7, \tag{5.2}$$
$$V_3 = -B_{34}V_4 - B_{35}V_5 - B_{36}V_6 - B_{37}E_7, \tag{5.3}$$
$$V_8 = -B_{84}V_4 - B_{85}V_5 - B_{86}V_6 - B_{87}E_7. \tag{5.4}$$

Asimismo, las corrientes de las ramas del árbol se expresan con las de enlace. El superíndice T indica transposición:

$$I_4 = B_{14}^T I_1 + B_{24}^T I_2 + B_{34}^T I_3 + B_{84}^T J_8, \tag{5.5}$$
$$I_5 = B_{25}^T I_2 + B_{35}^T I_3 + B_{85}^T J_8, \tag{5.6}$$
$$I_6 = B_{36}^T I_3 + B_{86}^T J_8, \tag{5.7}$$
$$I_7 = B_{17}^T I_1 + B_{27}^T I_2 + B_{37}^T I_3 + B_{87}^T J_8. \tag{5.8}$$

De (5.1) y (5.7) se obtienen también las derivadas:

$$\dot{V}_1 = -B_{14}\dot{V}_4 - B_{17}\dot{E}_7, \tag{5.9}$$
$$\dot{I}_6 = B_{36}^T \dot{I}_3 + B_{86}^T \dot{J}_8. \tag{5.10}$$

Estas últimas ecuaciones requieren las derivadas temporales de las fuentes, $\dot{E}_7$ y $\dot{J}_8$. Junto con las relaciones de los componentes, permiten derivar todas las tensiones y corrientes de la red a partir de las variables conocidas.

Magnitudes asociadas a resistencias

Son I_2, I_5, V_2 y V_5. La ley de Ohm da

$$V_2 = R_{22} I_2, \tag{5.11}$$
$$V_5 = R_{55} I_5. \tag{5.12}$$

Al combinar (5.2), (5.6), (5.11) y (5.12) resulta

$$I_2 = M_R^{-1}\{-B_{24}V_4 - B_{27}E_7 - B_{25}R_{55}[B_{35}^T I_3 + B_{85}^T J_8]\}, \tag{5.13}$$
$$M_R = R_{22} + B_{25}R_{55}B_{25}^T. \tag{5.14}$$

Por tanto, I_2 se determina directamente a partir de las variables de estado V_4 e I_3, las fuentes E_7 y J_8, y las submatrices pertinentes de B. Luego, (5.6), (5.11) y (5.12) proporcionan I_5, V_2 y V_5.

Como alternativa, esas mismas ecuaciones permiten calcular primero V_5:

$$V_5 = M_G^{-1}\{-B_{25}^T G_{22}[B_{24}V_4 + B_{27}E_7] + B_{35}^T I_3 + B_{85}^T J_8\}, \tag{5.15}$$
$$M_G = G_{55} + B_{25}^T G_{22} B_{25}.$$

Después se obtiene V_2 de (5.2), y $I_2 = G_{22}V_2$ e $I_5 = G_{55}V_5$. Los dos procedimientos difieren en el tamaño de la matriz que deben invertir: el orden de M_R es el número de resistencias de clase 2; el de M_G, el número de resistencias de clase 5. Con resistencias constantes basta una inversión y, en la práctica, la diferencia es pequeña. Si al menos una resistencia varía, hay que invertir una matriz en cada paso y su tamaño influye en el tiempo de cálculo. SCEPTRE elige automáticamente el procedimiento apropiado para cada red.

Magnitudes asociadas a capacitores

En cada paso se necesitan I_1, I_4, V_1 y V_4; la rutina de integración calcula V_4. En principio se cumple

$$\dot{V}_4 = S_{44} I_4, \tag{5.16}$$
$$\dot{V}_1 = S_{11} I_1. \tag{5.17}$$

De (5.5), (5.9), (5.16) y (5.17) se obtiene

$$I_1 = M_S^{-1}\{-B_{14}S_{44}[B_{24}^T I_2 + B_{34}^T I_3 + B_{84}^T J_8] - B_{17}\dot{E}_7\}, \tag{5.18}$$
$$M_S = S_{11} + B_{14}S_{44}B_{14}^T. \tag{5.19}$$

Así se conocen todas las magnitudes para calcular I_1. Las ecuaciones (5.5), (5.1) y (5.16) dan I_4, V_1 y la derivada de V_4, respectivamente. *Nota de traducción:* el original remite en este paso a (5.2), aunque la relación capacitiva pertinente es (5.17), y escribe b_{84} en (5.18), donde el resto de la notación emplea B_{84}.

Magnitudes asociadas a inductancias

Estas son $\dot{I}_3$, V_3, V_6 e I_6. Las relaciones básicas son

$$V_3 = L_{33}\dot{I}_3 + L_{36}\dot{I}_6, \tag{5.20}$$
$$V_6 = L_{63}\dot{I}_3 + L_{66}\dot{I}_6. \tag{5.21}$$

La combinación de (5.3), (5.10), (5.20) y (5.21) conduce a

$$\dot{I}_3 = M_L^{-1}\{-B_{34}V_4 - B_{35}V_5 - B_{37}E_7 - [B_{36}L_{66}B_{86}^T + L_{36}B_{86}^T]\dot{J}_8\}, \tag{5.22}$$
$$M_L = L_{33} + L_{36}B_{36}^T + B_{36}L_{63} + B_{36}L_{66}B_{36}^T. \tag{5.23}$$

Los vectores V_3, V_6 e I_6 se derivan, respectivamente, de (5.10) con (5.20), de (5.10) con (5.21), y de (5.7). Una vez conocidas las magnitudes anteriores, las corrientes de las fuentes de tensión y las tensiones de las fuentes de corriente se obtienen directamente de (5.8) y (5.4).

5.3.2. Esquema de solución

Las operaciones matriciales anteriores permiten calcular la evolución temporal de la red, pero deben repetirse en cada paso de integración. En ciertos casos puede reducirse el trabajo aprovechando la estructura de la matriz B.

Caso $B_{14} = B_{25} = B_{36} = 0$

Varias tensiones y corrientes se obtienen directamente de B. En general, las tensiones V_{enlace} y las corrientes $I_{\text{árbol}}$ cumplen

$$V_{\text{enlace}} = -B\,V_{\text{árbol}}, \tag{5.24}$$
$$I_{\text{árbol}} = B^T I_{\text{enlace}}. \tag{5.25}$$

Con el orden de la tabla 5.2, estas relaciones producen (5.1)–(5.8). Si $B_{25} = 0$, las ecuaciones (5.2) y (5.6) dan directamente V_2 e I_5. Conocidas las magnitudes resistivas, y si $B_{14} = 0$, (5.5) da I_4; de forma análoga, (5.3) da V_3 cuando $B_{36} = 0$. Por tanto, si las tres submatrices son nulas, V_2, I_5, I_4, V_3, I_6 y V_1 se obtienen por sustitución, sin inversiones de matrices. Las relaciones (5.11), (5.12), (5.18), (5.21), (5.16) y (5.22) proporcionan luego I_2, V_5, I_1, V_6, $\dot{V}_4$ y $\dot{I}_3$. SCEPTRE genera conforme a este esquema las instrucciones FORTRAN que se ejecutan en cada paso.

Nota de traducción: en (5.25) el original imprime B sin transponer; aquí se escribe B^T por coherencia con (5.5)–(5.8). Asimismo, donde el original enumera V_4 e I_3 como resultados de (5.16) y (5.22), estas ecuaciones calculan sus derivadas.

Caso $B_{25} \neq 0$

En redes grandes es habitual que B_{25} no sea nula. Entonces I_5 y V_2 no se calculan por simple sustitución: hay que utilizar (5.13) o (5.15), que incluyen una inversión de matriz. La figura 5.2 del original muestra un ejemplo con $B_{14} = B_{36} = 0$ y $B_{25} \neq 0$:

Las resistencias R_3 y R_D no contribuyen a B_{25}; por sustitución directa se obtiene

		Clase 4		Clase 5				Clase 7
		C_1	C_2	R_A	R_B	R_C	R_D	E_1
Clase 2	R_1	1	0	1	0	0	0	-1
	R_2	0	1	0	1	1	0	0
	R_3	0	1	0	0	0	0	-1
	R_4	0	0	1	0	0	0	0
Clase 3	L_1	0	0	0	0	0	1	0
	L_2	0	0	1	0	0	0	0

Figura 5.1: Ejemplo de matriz B con $B_{14} = B_{36} = 0$ y $B_{25} \neq 0$ (figura 5.2 del original).

$$V_{R3} = E_1 - V_{C2}, \qquad I_{RD} = I_{L1},$$
$$I_{R3} = \frac{E_1 - V_{C2}}{R_3}, \qquad V_{RD} = I_{L1} R_D.$$

Estas magnitudes se calculan en cada paso sin invertir matrices. Las demás magnitudes resistivas se obtienen mediante (5.13) y (5.14), y después se calculan las restantes variables de la red. Las únicas inversiones que deben repetirse en cada paso de integración son las necesarias para hallar I_{R1}, I_{R2} e I_{R4}.

5.3.3. Derivadas temporales de las fuentes

Las ecuaciones (5.9) y (5.10) necesitan $\dot{E}_7$ y $\dot{J}_8$. Bajo determinadas condiciones el usuario debe suministrar esas derivadas.

Fuente de tensión variable con $B_{17} \neq 0$

Si $B_{17} \neq 0$ y la fuente de tensión es constante, SCEPTRE fija su derivada temporal en cero. Si se necesita una derivada no nula y esta no está definida, la ejecución termina con un mensaje de error. Ocurre, por ejemplo, en mallas compuestas solo por capacitores y al menos una fuente de tensión variable.

Fuente de corriente variable con $B_{86} \neq 0$

Para fuentes de corriente y B_{86} rige la misma regla. El problema aparece en conjuntos de corte que contienen únicamente inductancias y al menos una fuente de corriente variable.

5.3.4. Fuentes linealmente dependientes

SCEPTRE permite definir fuentes de tensión y de corriente que dependen linealmente de tensiones y corrientes en resistencias. Un caso habitual es el circuito equivalente de pequeña señal de un transistor, representado en la figura 5.3 del original.

Estas fuentes requieren un tratamiento particular. Al comienzo de un paso de integración no se conocen directamente a partir de las variables de estado; usar el valor del paso previo produciría un *retardo de cálculo* (*computational delay*; véase el apartado 3.7). En la matriz B se incorporan para ello las clases Y y X:

strömen abhängig sind. Ein typisches Anwendungsg
torersatzschaltungen (siehe Abb. (5.3)).

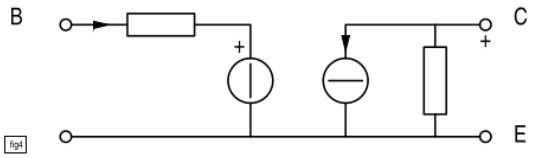


Figura 5.2: Circuito equivalente de pequeña señal con parámetros h (figura 5.3 del original).

$$B_{XY} = \begin{array}{c|ccccc} & 4 & 5 & 6 & 7 & Y \\ \hline 1 & B_{14} & 0 & 0 & B_{17} & B_{1Y} \\ 2 & B_{24} & B_{25} & 0 & B_{27} & B_{2Y} \\ 3 & B_{34} & B_{35} & B_{36} & B_{37} & B_{3Y} \\ 8 & B_{84} & B_{85} & B_{86} & B_{87} & B_{8Y} \\ X & B_{X4} & B_{X5} & B_{X6} & B_{X7} & B_{XY} \end{array}$$

Con estas clases se amplían (5.2) y (5.6):

$$V_2 = -B_{24}V_4 - B_{25}V_5 - B_{27}E_7 - B_{2Y}E_Y, \tag{5.26}$$
$$I_5 = B_{25}^T I_2 + B_{35}^T I_3 + B_{85}^T J_8 + B_{X5}^T J_X. \tag{5.27}$$

Cada fuente de tensión de clase Y depende de la tensión en alguna resistencia, ya sea del árbol o de enlace. Por eso

$$E_Y = k_1 V_2 + k_2 V_5. \tag{5.28}$$

k_1 y k_2 son matrices constantes. Ambas tienen tantas filas como fuentes de tensión de clase Y; k_1 tiene tantas columnas como resistencias de clase 2 y k_2, tantas como resistencias de clase 5. E_Y es el vector de fuentes de tensión de clase Y.

Análogamente, cada fuente de corriente de clase X depende de la corriente en una resistencia:

$$J_X = k_3 I_2 + k_4 I_5. \tag{5.29}$$

k_3 y k_4 son matrices constantes con tantas filas como fuentes de corriente de clase X; sus columnas corresponden, respectivamente, a las resistencias de clases 2 y 5. J_X es el vector de fuentes de corriente de clase X.

Sustituyendo (5.28), (5.29), (5.11) y (5.12) en (5.26) y (5.27), se llega a

$$R_{22}I_2 + B_{25}V_5 + B_{2Y}[k_1 R_{22} I_2 + k_2 V_5] = -B_{24}V_4 - B_{27}E_7, \tag{5.30}$$
$$G_{55}V_5 - B_{25}^T I_2 - B_{X5}^T[k_3 I_2 + k_4 G_{55} V_5] = B_{35}^T I_3 + B_{85}^T J_8. \tag{5.31}$$

En forma de bloques, el sistema es

$$\begin{bmatrix} R_{22} + B_{2Y}k_1 R_{22} & B_{25} + B_{2Y}k_2 \\ -B_{25}^T - B_{X5}^T k_3 & G_{55} - B_{X5}^T k_4 G_{55} \end{bmatrix} \begin{bmatrix} I_2 \\ V_5 \end{bmatrix} = \begin{bmatrix} -B_{24}V_4 - B_{27}E_7 \\ B_{35}^T I_3 + B_{85}^T J_8 \end{bmatrix}.$$

Si M_{RG} designa la matriz de bloques de la izquierda, I_2 y V_5 se calculan sin retardo:

$$\begin{bmatrix} I_2 \\ V_5 \end{bmatrix} = M_{RG}^{-1} \begin{bmatrix} -B_{24}V_4 - B_{27}E_7 \\ B_{35}^T I_3 + B_{85}^T J_8 \end{bmatrix}.$$

Como k_1–k_4 contienen solo constantes, basta invertir M_{RG} una vez si las resistencias son constantes. Si alguna resistencia varía, la inversión debe repetirse en cada paso.

Las clases adicionales amplían también (5.5), (5.3), (5.4) y (5.8):

$$I_4 = B_{14}^T I_1 + B_{24}^T I_2 + B_{34}^T I_3 + B_{84}^T J_8 + B_{X4}^T J_X, \tag{5.32}$$
$$V_3 = -B_{34}V_4 - B_{35}V_5 - B_{36}V_6 - B_{37}E_7 - B_{3Y}E_Y, \tag{5.33}$$
$$V_8 = -B_{84}V_4 - B_{85}V_5 - B_{86}V_6 - B_{87}E_7 - B_{8Y}E_Y, \tag{5.34}$$
$$I_7 = B_{17}^T I_1 + B_{27}^T I_2 + B_{37}^T I_3 + B_{87}^T J_8 + B_{X7}^T J_X. \tag{5.35}$$

Se agregan las relaciones

$$V_X = -B_{X4}V_4 - B_{X5}V_5 - B_{X7}E_7 - B_{XY}E_Y, \tag{5.36}$$
$$I_Y = B_{2Y}^T I_2 + B_{3Y}^T I_3 + B_{8Y}^T J_8 + B_{XY}^T J_X. \tag{5.37}$$

Existe una limitación: al extender (5.9) y (5.10) aparecen las derivadas de las fuentes dependientes,

$$\dot{V}_1 = -B_{14}\dot{V}_4 - B_{17}\dot{E}_7 - B_{1Y}\dot{E}_Y, \tag{5.38}$$
$$\dot{I}_6 = B_{36}^T \dot{I}_3 + B_{86}^T \dot{J}_8 + B_{X6}^T \dot{J}_X. \tag{5.39}$$

Si $B_{1Y} \neq 0$ o $B_{X6} \neq 0$, se necesitan $\dot{E}_Y$ o $\dot{J}_X$. Al derivar (5.28) y (5.29) se necesitarían también $\dot{V}_2$, $\dot{V}_5$, $\dot{I}_2$ y $\dot{I}_5$, que SCEPTRE no conoce en ese momento. Por eso comprueba esas submatrices y termina con un mensaje de error si alguna es no nula. $B_{1Y} \neq 0$ en una malla compuesta solo por capacitores y al menos una fuente de tensión controlada; $B_{X6} \neq 0$ en un conjunto de corte formado solo por inductancias y al menos una fuente de corriente controlada. *Nota de traducción:* la ecuación (5.38) del original imprime $R_{17}\dot{E}_7$; se utiliza $B_{17}\dot{E}_7$ por coherencia con (5.9) y el resto de la matriz B.

5.3.5. Métodos de integración

Para resolver las ecuaciones de estado, SCEPTRE ofrece tres métodos explícitos (`XPO`, `RUK` y `TRAP`) y uno implícito (`IMPLICIT`). Los métodos de Runge–Kutta y del trapecio se adaptaron, con modificaciones, de otros programas de redes. El método exponencial se desarrolló en el IBM Scientific Center de Palo Alto, California, y se adaptó a SCEPTRE. Según el original, casi siempre resulta más rápido que los demás. Si `RUN CONTROLS` no indica ningún método, se elige `XPO`. El método implícito es particularmente conveniente cuando conviven constantes de tiempo muy grandes y muy pequeñas.

Método de Runge–Kutta, `RUK`

Para el sistema diferencial $\dot{y} = f(t, y)$ se aplica el método de cuarto orden:

$$
\begin{aligned}
y(t+h) &= y(t) + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6},\\
k_1 &= hf[t, y(t)],\\
k_2 &= hf\left[t + \frac{h}{2}, y(t) + \frac{k_1}{2}\right],\\
k_3 &= hf\left[t + \frac{h}{2}, y(t) + \frac{k_2}{2}\right],\\
k_4 &= hf[t+h, y(t) + k_3].
\end{aligned}
$$

El control del paso variable utiliza además

$$k_5 = hf[t+h, y(t+h)], \qquad E = k_1 - 2k_3 - 2k_4 + 3k_5.$$

En un sistema de n variables se consideran los vectores $y(t) = [y_1(t), \ldots, y_n(t)]$ y $E = (e_1, \ldots, e_n)$. Para límites de error no negativos u_1, u_2, l_1, l_2, se definen, para cada $k = 1, \ldots, n$,

$$U_k = u_1 + u_2|y_k(t+h)|, \qquad L_k = l_1 + l_2|y_k(t+h)|.$$

El paso se ajusta con las reglas siguientes:

- Si $|e_k| > 1{,}5U_k$ para algún k, se reduce h a la mitad y se restablecen el tiempo, y y $\dot{y}$ a sus valores anteriores, en t.
- Si $|e_k| > 0{,}75U_k$ para algún k, pero $|e_k| \le 1{,}5U_k$ para todos, se conserva el paso actual y se reduce a la mitad el siguiente.
- Si $L_k \le |e_k| \le 0{,}75U_k$ para todos los componentes, el paso no cambia.
- Si $|e_k| \le U_k$ para todos y $|e_k| < L_k$ para al menos uno, el paso se duplica después de siete pasos de integración.

Después de reducir o duplicar el paso deben transcurrir al menos siete pasos de integración antes de que pueda duplicarse nuevamente. El original recomienda para límites de error absolutos $u_1 = 0{,}0075$, $u_2 = 0$, $l_1 = 0{,}005$ y $l_2 = 0$; para límites relativos, $u_1 = u_2 = l_1 = l_2 = 0{,}005$. Estos valores pueden establecerse mediante las órdenes `MAXIMUM/MINIMUM ABSOLUTE/RELATIVE ERROR` de `RUN CONTROLS`.

Método del trapecio modificado, `TRAP`

Método. Para $\dot{y}(t) = f(t, y(t))$, con $y(0) = y_0$, el original da el siguiente esquema predictor–corrector:

$$y_p\left(t + \frac{3h}{2}\right) = y(t) + \frac{h}{2}\dot{y}(t), \tag{5.40}$$

$$\dot{y}_p\left(t + \frac{3h}{2}\right) = f\left[t + \frac{3h}{2}, y_p\left(t + \frac{3h}{2}\right)\right] = f\left[t + \frac{3h}{2}, y(t) + \frac{h}{2}\dot{y}(t)\right], \tag{5.41}$$

$$y_c(t+h) = y(t) + \frac{h}{3}\left[2\dot{y}(t) + \dot{y}_p\left(t + \frac{3h}{2}\right)\right], \tag{5.42}$$

$$\dot{y}_c(t+h) = f[t+h, y_c(t+h)]. \tag{5.43}$$

El estimador empleado para ajustar el paso es

$$E = \frac{2h}{3}\left|\dot{y}_p\left(t+\frac{3h}{2}\right) - \dot{y}(t)\right|. \tag{5.44}$$

Error local. Si $\ddot{y} = f_t + \dot{y}f_y$, la solución exacta y_T se aproxima por

$$y_T(t+h) = y(t) + h\dot{y}(t) + \frac{h^2}{2}(f_t + \dot{y}f_y). \tag{5.45}$$

Una expansión de la evaluación del predictor da

$$f\left[t+\frac{3h}{2}, y(t)+\frac{h}{2}\dot{y}(t)\right] \doteq f(t,y(t)) + \frac{3h}{2}f_t + \frac{h}{2}\dot{y}f_y. \tag{5.46}$$

Sustituyendo en (5.42) se obtiene $y_c(t+h) \doteq y(t) + h\dot{y}(t) + \frac{h^2}{2}f_t + \frac{h^2}{6}\dot{y}f_y$. Por diferencia con (5.45), el error local aproximado es $y_T(t+h) - y_c(t+h) \doteq \frac{h^2}{3}\dot{y}f_y$. A partir de (5.41) y (5.46) también se obtiene

$$\frac{2h}{3}\left[\dot{y}_p\left(t+\frac{3h}{2}\right) - \dot{y}(t)\right] \doteq h^2 f_t + \frac{h^2}{3}\dot{y}f_y. \tag{5.47}$$

Si $f_t = 0$, (5.47) estima el error local al despreciar términos de orden superior. El original afirma que, para sistemas de la forma $\dot{y} = A(y)y + B$ con A lineal por tramos, este estimador y su control de paso han resultado suficientes.

Estabilidad. Para $\dot{y} = -ay$, $y(0) = y_0$ y $a > 0$, el esquema (5.40)–(5.43) produce

$$y_c(nh) = \left(1 - ah + \frac{(ah)^2}{6}\right)^n y(0).$$

Se demuestra por inducción que la condición necesaria y suficiente de estabilidad numérica es

$$\left|1 - ah + \frac{(ah)^2}{6}\right| < 1 \quad \Longleftrightarrow \quad 0 < ah < 6.$$

Así, el radio de estabilidad es $r = 6$. La figura 5.5 del original representa la condición sobre ah.

Las ecuaciones (5.40) y (5.41) requieren dos evaluaciones de derivadas por paso; por tanto, $p = 2$. El cociente r/p mide la eficiencia según el número mínimo de pasos por intervalo de tiempo: cuanto mayor es, menos pasos exige un a dado. El original compara varios métodos:

Método	r	p	r/p
Trapecio modificado	6	2	3
Euler [9]	2	1	2
Trapecio [10]	2	2	1
NIDE, un paso [11]	0,8	1	0,8
Runge–Kutta de cuarto orden	2,78	4	0,7
NIDE, dos pasos	1	2	0,5
Hamming [12]	0,85	2	0,425

Tabla 5.2: Comparación de r/p entre integradores (tabla 5.3 del original).

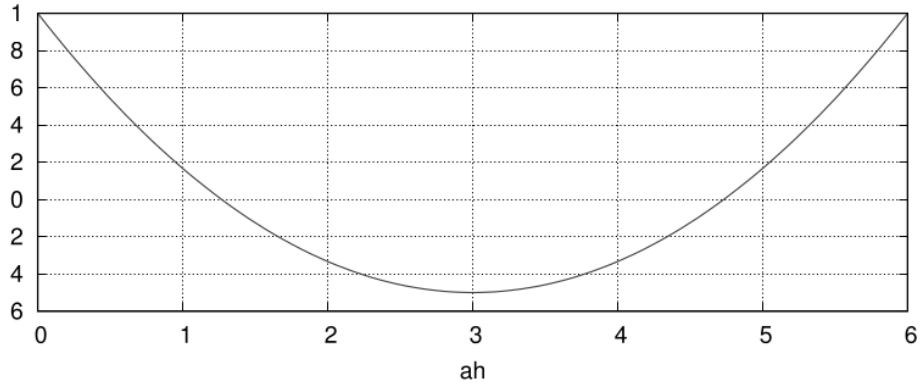


Figura 5.3: Condición de estabilidad $0 < ah < 6$ para `TRAP` (figura 5.5 del original).

Control del paso. Se emplea el estimador (5.44), denominado E_2 en la explicación del original, con los límites $U = u_1 + u_2|y_c(t+h)|$ y $L = l_1 + l_2|y_c(t+h)|$, donde $u_1, u_2, l_1, l_2 \geq 0$:

- Si $|E_2| > 0{,}75U$, se reduce el paso.
- Si $|E_2| < L$, se aumenta el paso.
- Si $L \leq |E_2| \leq 0{,}75U$, no se cambia el paso.

Como E_2 aproxima el error local, el original recomienda para control absoluto $u_1 = 10^{-2}$, $l_1 = 5 \cdot 10^{-4}$, $u_2 = l_2 = 0$; para control relativo, $u_1 = 10^{-3}$, $l_1 = 5 \cdot 10^{-5}$, $u_2 = 10^{-2}$ y $l_2 = 5 \cdot 10^{-4}$. Para reducir el paso a la mitad, los valores anteriores deben disminuirse por un factor de cuatro.

Método exponencial, `XPO`

Método. Para $\dot{y}(t) = f[t, y(t)]$, $y(0) = y_0$, el método utiliza, salvo dos excepciones descritas más adelante,

$$y(t+h) = y(t) + h\dot{y}_A(t) + \frac{e^{\lambda h} - 1}{\lambda h}\, h\dot{y}_p(t). \tag{5.48}$$

En esta expresión, $\dot{y}_A(t) = [y(t) - y(t - h_0)]/h_0$, donde $t - h_0$ es el instante anterior de integración, y

$$\dot{y}_p(t) = \dot{y}(t) - \dot{y}_A(t), \tag{5.49}$$

$$\lambda = \frac{\ddot{y}_p(t)}{\dot{y}_p(t)}. \tag{5.50}$$

Como el programa no calcula explícitamente $\ddot{y}(t)$, emplea la siguiente aproximación por un incremento $0 < \delta \leq h/4$:

$$\begin{aligned} y_\delta(t+\delta) &= y(t) + \delta\dot{y}(t), \\ \dot{y}_\delta(t+\delta) &= f[t+\delta, y_\delta(t+\delta)], \\ \ddot{y}_\delta(t) &= \frac{\dot{y}_\delta(t+\delta) - \dot{y}(t)}{\delta}. \end{aligned} \tag{5.51}$$

La primera excepción ocurre cuando $\ddot{y}_\delta(t)$ y $\dot{y}_p(t)$ tienen el mismo signo. Se fija entonces $\dot{y}_A(t) = 0$ y se usan

$$\dot{y}_p(t) = \dot{y}(t), \tag{5.52}$$

$$\lambda = \frac{\ddot{y}_\delta(t)}{\dot{y}(t)}. \tag{5.53}$$

De este modo se obtiene una mejor aproximación para $\ddot{y} = \lambda y + b$ sin degradar la calidad general del método. La segunda excepción se da si λ, calculada según (5.50) o (5.53), no es negativa. En ese caso se sustituye $(e^{\lambda h} - 1)/(\lambda h)$ por $1 + \lambda h/2$:

$$y(t+h) = y(t) + h\dot{y}_A(t) + \left(1 + \frac{\lambda h}{2}\right) h\dot{y}_p(t). \tag{5.54}$$

Error local. La serie de Taylor $(e^x - 1)/x = 1 + x/2 + x^2/6 + O(x^3)$ aplicada a (5.48) da

$$\begin{aligned} y(t+h) &= y(t) + h\dot{y}_A(t) + h\dot{y}_p(t) + \frac{h^2}{2}\lambda\dot{y}_p(t) + \frac{h^3}{6}\lambda^2\dot{y}_p(t) + O(h^4) \\ &= y(t) + h\dot{y}(t) + \frac{h^2}{2}\ddot{y}_\delta(t) + \frac{h^3}{6}\lambda\ddot{y}_\delta(t) + O(h^4). \end{aligned} \tag{5.55}$$

Para la aproximación (5.51) se escribe

$$\ddot{y}_\delta(t) = \ddot{y}(t) + \frac{\delta}{2}\left[\frac{d^3y}{dt^3} - \ddot{y}(t)\frac{\partial f}{\partial y}\right] + O(\delta^2). \tag{5.56}$$

Sustituyendo en (5.55), el original obtiene

$$y(t+h) = y(t) + h\dot{y}(t) + \frac{h^2}{2}\ddot{y}(t) + \frac{h^2\delta}{4}\left[\frac{d^3y}{dt^3} - \ddot{y}(t)\frac{\partial f}{\partial y}\right] + O(h^3). \tag{5.57}$$

Como $\delta \leq h/4$, el error es de orden $O(h^3)$.

Control del paso. Sea $y_T(t+\xi)$ la solución exacta y $y_c(t+\xi)$ la calculada, con $0 \leq \xi \leq h$. La derivada de la expresión exponencial calculada, $\dot{y}_e$, es

$$\dot{y}_e(t+\xi) = \begin{cases} \dot{y}_A(t) + (1+\lambda\xi)\dot{y}_p(t), & \lambda \geq 0, \\ \dot{y}_A(t) + e^{\lambda\xi}\dot{y}_p(t), & \lambda < 0. \end{cases}$$

Se denomina $\dot{y}_c(t+\xi)$ a la derivada de $y_c(t+\xi)$. Idealmente el paso se elegiría a partir de $E_0 = \int_0^h [\dot{y}_T(t+\xi) - \dot{y}_e(t+\xi)]\, d\xi$, pero $\dot{y}_T$ no se conoce salvo en $\xi = 0$. Se aprovecha que

$$\begin{aligned} \dot{y}_c(t+\xi) &= f\{t+\xi, y_T(t+\xi) + [y_c(t+\xi) - y_T(t+\xi)]\} \\ &= \dot{y}_T(t+\xi) + O(\xi^3), \end{aligned} \tag{5.58}$$

de donde

$$E_0 = \int_0^h [\dot{y}_c(t+\xi) - \dot{y}_e(t+\xi)]\, d\xi + \int_0^h [\dot{y}_T(t+\xi) - \dot{y}_c(t+\xi)]\, d\xi, \tag{5.59}$$

$$= \int_0^h [\dot{y}_c(t+\xi) - \dot{y}_e(t+\xi)]\, d\xi + O(h^4). \tag{5.60}$$

Si $\ddot{y}_c(t+\xi) - \ddot{y}_e(t+\xi)$ no cambia de signo para $0 \leq \xi \leq h$, entonces

$$|E_0| \leq h|\dot{y}_T(t+h) - \dot{y}_e(t+h)| \leq h|\dot{y}_c(t+h) - \dot{y}_e(t+h)| + |O(h^4)|. \tag{5.61}$$

Para h pequeño se toma como estimador $E = h|\dot{y}_c(t+h) - \dot{y}_e(t+h)|$. Con límites de error no negativos u_1, u_2, l_1, l_2, se fijan $U = u_1 + u_2|y_c(t+h)|$ y $L = l_1 + l_2|y_c(t+h)|$:

- Si $|E| > U$, se reduce el paso.
- Si $|E| < L$, se aumenta el paso.
- Si $L \leq |E| \leq U$, el paso no cambia.

El original recomienda para límites absolutos $u_1 = 75 \cdot 10^{-3}$, $l_1 = 2 \cdot 10^{-4}$ y $u_2 = l_2 = 0$; para límites relativos, $u_1 = u_2 = 5 \cdot 10^{-3}$, $l_1 = 10^{-4}$ y $l_2 = 2 \cdot 10^{-4}$. *Nota de traducción:* el integral inicial de E_0 en el original contiene $\dot{y}_e(t-\xi)$, mientras que su desarrollo usa $t+\xi$; aquí se emplea este último. Su tercera regla de control imprime $L \geq E \geq U$, imposible para los límites usuales $L < U$; se expresa como $L \leq |E| \leq U$.

Límites de error absolutos y relativos

Los tres métodos explícitos anteriores aproximan las variables de estado en cada instante de integración y controlan automáticamente el tamaño del paso. Comparan una magnitud de control, relacionada con el paso y las derivadas, con un límite de error que puede ser constante (absoluto) o depender de la magnitud de las variables de estado (relativo). Si las variables tienen escalas muy distintas, imponer la misma precisión absoluta a todas puede exigir pasos innecesariamente pequeños para las de mayor valor. Los límites relativos evitan este inconveniente; los absolutos son más apropiados cuando todas las variables de estado son pequeñas.

Los tres métodos explícitos emplean por defecto límites relativos, aunque el usuario también puede establecer límites absolutos en `RUN CONTROLS`. Valores grandes de los límites superiores u_1 o u_2 reducen la probabilidad de que el paso se divida por dos; al aumentar los límites inferiores l_1 o l_2, aumenta la probabilidad de que el paso crezca. En general, exigir más precisión aumenta el tiempo de cálculo.

Método implícito, `IMPLICIT`

Formulación general. La expresión general que presenta el original para los métodos implícitos es

$$Y(n+1) = \sum_{i=0}^{p} a_i Y(n-i) + h \sum_{i=-1}^{p} b_i \dot{Y}(n-i). \tag{5.62}$$

Y es el vector de variables de estado; $\dot{Y}$, su derivada; n, el número de paso; h, el tamaño del paso, y a_i, b_i son constantes. Si $p > 0$, el esquema es multipaso. El método de Euler implícito es el caso sencillo

$$Y(n+1) = Y(n) + h\dot{Y}(n+1). \tag{5.63}$$

Nota de traducción: en la segunda suma de (5.62) el original imprime $\dot{Y}(n-1)$ para todos los índices i; se representa $\dot{Y}(n-i)$ porque de otro modo la suma no describe los distintos pasos y no incluye el caso (5.63) con $i = -1$.

Para un sistema de m ecuaciones, la iteración implícita se expresa de forma compacta como

$$\left[I - hg\, J(Y,t)\right] \Delta Y^k = -F(Y,t), \qquad J_{ij}(Y,t) = \frac{\partial \dot{Y}_i}{\partial Y_j}, \quad 1 \le i,j \le m. \tag{5.64}$$

Esta es la misma matriz que el original despliega elemento por elemento: su entrada diagonal i es $1 - hg\,\partial\dot{Y}_i/\partial Y_i$ y una entrada fuera de la diagonal es $-hg\,\partial\dot{Y}_i/\partial Y_j$. Se repite hasta converger; k cuenta las aproximaciones, g depende del orden del método y F_i depende de la ecuación diferencial i, del paso y de las m variables de estado. Para problemas grandes se recurre a matrices dispersas.

Matriz jacobiana. SCEPTRE conoce las derivadas temporales $\dot{Y}_i$, pero debe calcular las parciales $\partial\dot{Y}_i/\partial Y_j$. El original parte de la forma matricial

$$\dot{Y} = AY + BU + N\dot{U}. \tag{5.65}$$

La matriz A reúne las derivadas parciales que se necesitan en (5.64). A continuación se deduce su forma simbólica a partir de las ecuaciones de la red.

Al sustituir (5.9) y (5.13) en (5.5), resulta

$$\begin{aligned} C_{44}\dot{V}_4 = {} & B_{14}^T C_{11}(-B_{14}\dot{V}_4 - B_{17}\dot{E}_7) \\ & + B_{24}^T M_R^{-1}(-B_{24}V_4 - B_{27}E_7 - B_{25}R_{55}B_{35}^T I_3 - B_{25}R_{55}B_{85}^T J_8) \\ & + B_{34}^T I_3 + B_{84}^T J_8. \end{aligned} \tag{5.66}$$

Reordenando los términos se obtiene

$$\begin{aligned} (C_{44} + B_{14}^T C_{11} B_{14})\dot{V}_4 = {} & - B_{14}^T C_{11} B_{17}\dot{E}_7 \\ & - B_{24}^T M_R^{-1}(B_{24}V_4 + B_{27}E_7 + B_{25}R_{55}B_{35}^T I_3 + B_{25}R_{55}B_{85}^T J_8) \\ & + B_{34}^T I_3 + B_{84}^T J_8. \end{aligned} \tag{5.67}$$

Por otro lado, la sustitución de (5.21), (5.10), (5.20) y (5.22) en (5.3) da

$$\begin{aligned} L_{33}\dot{I}_3 + L_{36}(B_{36}^T\dot{I}_3 + B_{86}^T\dot{J}_8) = {} & - B_{34}V_4 \\ & - B_{35}M_G^{-1}\{-B_{25}^T G_{22}(B_{24}V_4 + B_{27}E_7) + B_{35}^T I_3 + B_{85}^T J_8\} \\ & - B_{36}(L_{63} + L_{66}B_{36}^T)\dot{I}_3 - B_{36}L_{66}B_{86}^T\dot{J}_8 - B_{37}E_7. \end{aligned} \tag{5.68}$$

Tras reagrupar,

$$\begin{aligned} [L_{33} + L_{36}B_{36}^T + B_{36}L_{63} + B_{36}L_{66}B_{36}^T]\dot{I}_3 = {} & - B_{34}V_4 \\ & - B_{35}M_G^{-1}(-B_{25}^T G_{22}B_{24}V_4 - B_{25}^T G_{22}B_{27}E_7 + B_{35}^T I_3 + B_{85}^T J_8) \\ & - B_{37}E_7 - L_{36}B_{86}^T\dot{J}_8 - B_{36}L_{66}B_{86}^T\dot{J}_8. \end{aligned} \tag{5.69}$$

Al tomar de (5.67) y (5.69) solo los coeficientes de V_4, $\dot{V}_4$, I_3 y $\dot{I}_3$, se obtiene la parte de la matriz A que corresponde a las variables de estado:

$$\begin{bmatrix} \dot{V}_4 \\ \dot{I}_3 \end{bmatrix} = \begin{bmatrix} M_C & 0 \\ 0 & M_L \end{bmatrix}^{-1} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} V_4 \\ I_3 \end{bmatrix}, \tag{5.70}$$

donde $M_C = C_{44} + B_{14}^T C_{11} B_{14}$, $M_L = L_{33} + L_{36}B_{36}^T + B_{36}L_{63} + B_{36}L_{66}B_{36}^T$ y los bloques del original se han abreviado como

$$\begin{aligned} A_{11} &= -B_{24}^T M_R^{-1} B_{24}, \\ A_{12} &= B_{34}^T - B_{24}^T M_R^{-1} B_{25} R_{55} B_{35}^T, \\ A_{21} &= B_{35} M_G^{-1} B_{25}^T G_{22} B_{24} - B_{34}, \\ A_{22} &= -B_{35} M_G^{-1} B_{35}^T. \end{aligned}$$

Esta matriz tiene la forma del primer término a la derecha de (5.65). SCEPTRE también puede obtener la jacobiana numéricamente: llama m veces a la subrutina `SIMUL8`, perturbando en cada llamada una variable de estado,

$$\dot{Y}^{(j)} = F(Y_1, \ldots, Y_{j-1}, Y_j + \Delta Y_j, Y_{j+1}, \ldots, Y_m, t),$$

y aproxima la derivada parcial mediante $\partial \dot{Y}_i / \partial Y_j \approx [\dot{Y}_i^{(j)} - \dot{Y}_i]/\Delta Y_j$.

5.4. Análisis de corriente continua (DC)

SCEPTRE ofrece cinco clases de análisis de corriente continua: condiciones iniciales, sensibilidad, Monte Carlo, peor caso y optimización. Cada uno puede usarse de manera autónoma o aportar valores iniciales a un análisis transitorio o de corriente alterna. También es posible que el usuario proporcione directamente esas condiciones. Los cinco análisis parten del cálculo de condiciones iniciales y aprovechan sus resultados. Para obtenerlas puede emplearse Newton–Raphson o el método implícito.

5.4.1. Método de Newton–Raphson

Procedimiento

En muchos circuitos, las condiciones iniciales determinan de manera decisiva la evolución transitoria y deben calcularse en el instante inicial t_0. Una posibilidad consiste en realizar un análisis transitorio adicional con todas las fuentes mantenidas constantes en su valor de t_0. Otra es iterar sin tomar el tiempo como variable independiente; cuando la red converge, este procedimiento suele ahorrar tiempo de cálculo. Es independiente de los métodos empleados durante el análisis transitorio.

Se considera un *árbol L*, que ordena preferentemente los elementos E, L, R y C. La tabla 5.4 del original presenta su matriz B general. Algunas submatrices son nulas por construcción; por ejemplo, $B_{24} = 0$, pues el orden de preferencia impide una malla formada por una resistencia de enlace y un capacitor del árbol.

$B_{\mathrm{L}} =$

	4	5	6	7
1	B_{14}	B_{15}	B_{16}	B_{17}
2	0	B_{25}	B_{26}	B_{27}
3	0	0	B_{36}	B_{37}
8	B_{84}	B_{85}	B_{86}	B_{87}
9	B_{94}	B_{95}	B_{96}	B_{97}
0	B_{04}	B_{05}	B_{06}	B_{07}

Suponiendo nulos V_6, I_1, I_4 y B_{94}, se deduce

$$V_9 + B_{95}V_5 + B_{97}E_7 = 0, \tag{5.71}$$
$$V_2 + B_{25}V_5 + B_{27}E_7 = 0, \tag{5.72}$$
$$I_5 - B_{25}^T I_2 - B_{85}^T J_8 - B_{95}^T J_9 - B_{05}^T J_0 = 0. \tag{5.73}$$

Para expresar las variables mediante V_9, V_2, V_5 y las fuentes, se sustituyen $I_5 = G_{55}V_5$, $I_2 = G_{22}V_2$, $J_0 = \alpha J_9$ y $J_9 = G_{99}V_9 + Q_9$. Aquí α tiene tantas filas como fuentes de corriente de segundo orden y tantas columnas como fuentes de primer orden; G_{55} y G_{22} son diagonales de conductancias; G_{99} contiene las conductancias de diodos y transistores. Los términos Q se describen en el anexo D.4. *Nota de traducción:* la sustitución de I_2 aparece defectuosa en el original como $G_{22}V_5 r$; (5.74) muestra que corresponde $G_{22}V_2$.

Así, (5.73) se convierte en

$$G_{55}V_5 - B_{25}^T G_{22}V_2 - B_{85}^T J_8 - [B_{95}^T + B_{05}^T\alpha](G_{99}V_9 + Q_9) = 0. \tag{5.74}$$

Se denominan F_1, F_2 y F_3 los miembros izquierdos de (5.71), (5.72) y (5.74), respectivamente, todos dependientes de (V_9, V_2, V_5). Newton–Raphson linealiza estas funciones alrededor de los valores de la iteración actual. En notación vectorial,

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} + Z \begin{bmatrix} \Delta V_9 \\ \Delta V_2 \\ \Delta V_5 \end{bmatrix} = 0, \tag{5.75}$$

donde la jacobiana está formada por las derivadas parciales

$$Z = \begin{bmatrix} \partial F_1/\partial V_9 & \partial F_1/\partial V_2 & \partial F_1/\partial V_5 \\ \partial F_2/\partial V_9 & \partial F_2/\partial V_2 & \partial F_2/\partial V_5 \\ \partial F_3/\partial V_9 & \partial F_3/\partial V_2 & \partial F_3/\partial V_5 \end{bmatrix}. \tag{5.76}$$

Con n como índice de iteración y $\Delta V_j = V_j(n+1) - V_j(n)$ para $j = 9, 2, 5$, (5.75) equivale a

$$F(V(n)) + Z[V(n+1) - V(n)] = 0, \tag{5.77}$$

y, por tanto,

$$V(n+1) = V(n) - Z^{-1}F(V(n)), \qquad V = \begin{bmatrix} V_9 \\ V_2 \\ V_5 \end{bmatrix}. \tag{5.78}$$

La diferenciación de (5.71), (5.72) y (5.74) da explícitamente

$$Z = \begin{bmatrix} I & 0 & B_{95} \\ 0 & I & B_{25} \\ -[B_{95}^T + B_{05}^T\alpha]G_{99} & -B_{25}^T G_{22} & G_{55} \end{bmatrix}.$$

Al introducir esta matriz explícita en (5.78) se tiene, sin cambiar la regla de actualización,

$$V(n+1) = V(n) - \begin{bmatrix} I & 0 & B_{95} \\ 0 & I & B_{25} \\ -[B_{95}^T + B_{05}^T\alpha]G_{99} & -B_{25}^T G_{22} & G_{55} \end{bmatrix}^{-1} F(V(n)). \tag{5.79}$$

En la forma abreviada del original (véase el anexo D.7), la iteración puede escribirse

$$\begin{bmatrix} V_9(n+1) \\ V_2(n+1) \\ V_5(n+1) \end{bmatrix} = Z^{-1} \begin{bmatrix} -B_{97}E_7 \\ -B_{27}E_7 \\ [B_{95}^T + B_{05}^T\alpha]Q_9 + B_{85}^T J_8 \end{bmatrix}. \tag{5.80}$$

Se comienza con valores arbitrarios de V_9, V_2 y V_5, normalmente cero. En cada iteración se actualizan los términos del lado derecho de (5.80) que dependen de la tensión, se resuelve el sistema y se comprueba la convergencia de todas las tensiones. El original da el criterio

$$|V_{n+1} - V_n| \leq |V_{n+1}| + 0{,}0001. \tag{5.81}$$

Se admiten hasta cien iteraciones; si no se alcanza la convergencia se imprime un mensaje. Según la experiencia recogida en el manual, la mayoría de las redes convergen en menos de treinta. La convergencia empeora si una unión de diodo o transistor presenta una tensión directa elevada y se modela como $J = I_s(e^{\Theta V} - 1)$: la pendiente $G_J = dJ/dV = \Theta I_s e^{\Theta V}$ vuelve muy grande un término de G_{99} y, con ello, de Z, lo que exige incrementos muy pequeños de las variables independientes y muchas iteraciones. *Nota de traducción:* el criterio (5.81) se reproduce tal como está impreso; el original no indica un coeficiente relativo delante de $|V_{n+1}|$.

Una vez conocidos V_9, V_2 y V_5, se calculan las tensiones en capacitores y las corrientes en inductancias. En corriente continua, las corrientes de los capacitores y las tensiones de las inductancias deben ser cero.

Cálculo de las tensiones de los capacitores

La matriz del árbol L expresa las tensiones de capacitores en ramas de enlace mediante las tensiones de ramas del árbol:

$$V_1 = -B_{14}V_4 - B_{15}V_5 - B_{17}E_7. \tag{5.82}$$

La conservación de la carga eléctrica impone

$$-B_{14}^T C_{11} V_1 + C_{44} V_4 = -B_{14}^T C_{11} V_1(0) + C_{44} V_4(0), \tag{5.83}$$

donde $V_1(0)$ y $V_4(0)$ son tensiones iniciales indicadas por el usuario. Al combinar (5.82) y (5.83) se obtiene

$$V_4 = M_C^{-1}\{-B_{14}^T C_{11} B_{15} V_5 - B_{14}^T C_{11} B_{17} E_7 - B_{14}^T C_{11} V_1(0) + C_{44} V_4(0)\}, \tag{5.84}$$

con $M_C = B_{14}^T C_{11} B_{14} + C_{44}$. Conocido V_4, se calcula V_1 mediante (5.82). Un capacitor de clase 4 solo puede aparecer si hay un conjunto de corte capacitivo; de lo contrario no hace falta (5.84). La conservación de la carga permite tratar redes cuyas tensiones finales dependen de la carga inicial. En el ejemplo de la figura 5.6, con $C_1 = 9\,\text{pF}$ y $C_2 = 1\,\text{pF}$, si ambos capacitores comienzan descargados, al cerrar el interruptor se llega a $V_{C1} = 1\,\text{V}$ y $V_{C2} = 9\,\text{V}$. Si C_1 empieza con 5 V, las tensiones finales son 5,5 V y 4,5 V, respectivamente.

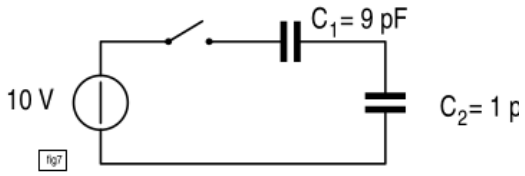


Figura 5.4: Circuito con conjunto de corte capacitivo (figura 5.6 del original).

Cálculo de las corrientes de las inductancias

Como V_2 ya se conoce, se calcula

$$I_2 = R_{22}^{-1} V_2. \tag{5.85}$$

La matriz B da las corrientes de las ramas del árbol:

$$I_6 = B_{26}^T I_2 + B_{36}^T I_3 + B_{86}^T J_8 + B_{96}^T J_9 + B_{06}^T J_0. \tag{5.86}$$

Para las corrientes iniciales $I_6(0)$ e $I_3(0)$, el original aplica la conservación de la energía magnética:

$$B_{36} L_{66} I_6 + L_{33} I_3 = B_{36} L_{66} I_6(0) + L_{33} I_3(0). \tag{5.87}$$

De (5.86) y (5.87) resulta

$$I_3 = M_l^{-1}\{L_{33} I_3(0) + B_{36} L_{66} I_6(0) - B_{36} L_{66}[B_{26}^T I_2 + B_{86}^T J_8 + B_{96}^T J_9 + B_{06}^T J_0]\}, \tag{5.88}$$

con $M_l = B_{36} L_{66} B_{36}^T + L_{33}$. Después, (5.86) proporciona I_6. Las inductancias de enlace, de clase 3, aparecen solo si hay un conjunto de corte inductivo; si no lo hay, basta (5.86). La ecuación (5.88) permite resolver redes cuyas corrientes finales dependen de los valores de las inductancias. En la figura 5.7, dos inductancias inicialmente sin corriente conducen, al cerrar el interruptor, $I_{L1} = 1\,\mathrm{A}$ e $I_{L2} = 9\,\mathrm{A}$.

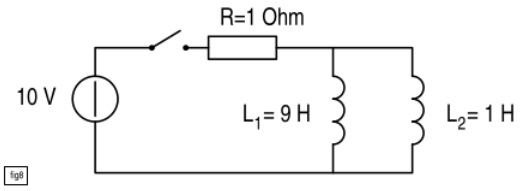


Figura 5.5: Circuito con una malla de inductancias (figura 5.7 del original).

Restricciones

El cálculo de condiciones iniciales está sujeto a restricciones que no necesariamente se aplican al análisis transitorio puro:

- No se permiten mallas formadas solo por fuentes de tensión e inductancias; implicarían una corriente infinita en la inductancia.
- No se permiten nodos conectados exclusivamente por fuentes de corriente y capacitores, como el nodo A de la figura 5.8. De lo contrario se viola (5.71) y Newton–Raphson tendría que incluir también V_4. Una resistencia grande entre A y tierra evita el problema. Esta restricción no rige para un análisis exclusivamente transitorio.
- Las corrientes de inductancias o resistencias no pueden ser variables independientes. Una tensión de capacitor solo puede serlo si hay una resistencia en paralelo, $V_C = V_R$, o una fuente de corriente en paralelo, $V_C = V_J$.
- Un conjunto de nodos unidos únicamente por capacitores solo se admite cuando todos esos capacitores son constantes.

- Una red formada por mallas que contienen solo inductancias se admite únicamente cuando todas las inductancias son constantes.

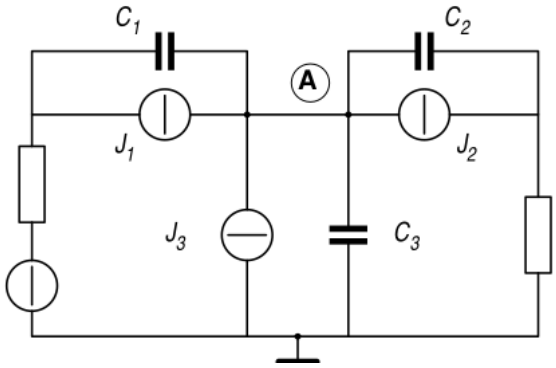


Figura 5.6: Nodo conectado por capacitores y fuentes de corriente (figura 5.8 del original).

5.4.2. Monte Carlo

El análisis de Monte Carlo repite el cálculo de condiciones iniciales mientras varían, dentro de tolerancias especificadas, los valores de ciertos elementos de la red. Estas desviaciones respecto del nominal, que pueden deberse por ejemplo a la fabricación, provocan variaciones en el comportamiento del circuito. SCEPTRE entrega estadísticas para las magnitudes seleccionadas por el usuario. Cada elemento con tolerancias recibe valores aleatorios dentro de sus límites, con distribución uniforme o normal gaussiana.

Como el cálculo se realiza en $t = 0$, el original señala que capacitores e inductancias no influyen en el resultado de este análisis. Se pueden variar resistencias, fuentes independientes de corriente o tensión, y fuentes de corriente controladas de primer y segundo orden mediante parámetros definidos.

Si μ es el valor medio, σ la desviación y ζ un número aleatorio de media cero y desviación estándar uno tomado de la distribución elegida, el valor del elemento es

$$\text{valor del elemento} = \mu + \zeta\sigma.$$

Cuando la distribución uniforme o la normal no sirven, pueden definirse parámetros para construir distribuciones más complejas mediante expresiones aritméticas y tablas. En cada corrida se asignan valores a los elementos con tolerancia y se calculan las condiciones iniciales. Para cada salida se informan el valor nominal, los extremos observados, la media y la desviación estándar. Si hay N corridas y las salidas son x_i, estas últimas se calculan como

$$M_x = \frac{\sum_{i=1}^{N} x_i}{N}, \qquad S_x = \sqrt{\frac{\sum_{i=1}^{N} (x_i - M_x)^2}{N-1}}.$$

5.4.3. Sensibilidad

El análisis de sensibilidad calcula cómo cambia una función de la red H con una variable independiente X. Informa tanto la sensibilidad no normalizada, $\partial H/\partial X$, como la normalizada, $(X/H)(\partial H/\partial X)$.

Primero se calculan las condiciones iniciales de la red original. Después se construye una red *adjunta* y se calculan en ella las condiciones iniciales para cada variable solicitada. Los productos apropiados entre corrientes y tensiones de ambas redes dan la sensibilidad. En la tabla 5.5, el subíndice a distingue las magnitudes de la red adjunta:

Variable independiente	**Sensibilidad**
Resistencia R	Producto de las corrientes por la misma resistencia en ambas redes: $I_R I_{Ra}$.
Fuente de tensión independiente E_7	Opuesto de la corriente por la misma fuente en la red adjunta: $-I_{E7a}$.
Fuente de corriente independiente J_8	Opuesto de la tensión en la misma fuente de la red adjunta: $-V_{J8a}$.
Fuente de corriente de segundo orden $J_0 = PJ_9$, con parámetro definido P	Corriente de la fuente de primer orden en la red original multiplicada por el opuesto de la tensión en la fuente de segundo orden adjunta: $-J_9 V_{J0a}$.

Tabla 5.3: Reglas de sensibilidad (tabla 5.5 del original).

5.4.4. Análisis del peor caso

Para determinar el caso más desfavorable, SCEPTRE extrapola linealmente el gradiente de una función objetivo F desde los valores nominales hasta los límites de tolerancia de una o varias variables independientes. El gradiente ∇F, obtenido del análisis nominal de la red, señala la dirección de mayor crecimiento de F. El procedimiento supone que la dependencia de F con las N variables independientes es lineal en la región considerada.

Si Y reúne los valores nominales, los vectores que llevan, respectivamente, al máximo y al mínimo estimados se expresan como

$$X_H = Y + \lambda \nabla F, \qquad \lambda > 0, \tag{5.89}$$

$$X_L = Y - \mu \nabla F, \qquad \mu > 0. \tag{5.90}$$

Se eligen λ y μ tan grandes como sea posible sin salir de los intervalos $P_i \le X_i \le Q_i$ para $i = 1, \ldots, N$. Si $\partial F / \partial X_i > 0$, al aumentar X_i desde Y_i hacia Q_i crece F; si la derivada es negativa, al disminuir X_i hacia P_i también crece F.

Una recta que atraviesa Y en la dirección del vector unitario u cumple

$$\frac{X_1 - Y_1}{u_1} = \frac{X_2 - Y_2}{u_2} = \cdots = \frac{X_N - Y_N}{u_N} = \text{constante}. \tag{5.91}$$

Cuando su componente j llega al límite inferior P_j, las otras componentes satisfacen $(X_i - Y_i)/u_i = (P_j - Y_j)/u_j$; la distancia recorrida es $D_{P_j} = |(P_j - Y_j)/u_j|$. Para el límite superior se obtiene análogamente $D_{Q_j} = |(Q_j - Y_j)/u_j|$. Entre las distancias que corresponden a los límites alcanzables en el sentido de ∇F, la menor determina D_{high}: es la máxima distancia recorrible antes de tocar una cara de la caja de tolerancias. En el sentido opuesto se obtiene D_{low}. La variable que limita un sentido puede ser distinta de la que limita el otro, como ilustra la figura 5.9.

Con esas distancias, los vectores de variables independientes son $X_H = Y + uD_{\text{high}}$ y $X_L = Y - uD_{\text{low}}$, en correspondencia con (5.89) y (5.90) porque u es proporcional al gradiente. Entonces se calculan la función objetivo y el gradiente en ambos puntos. Los

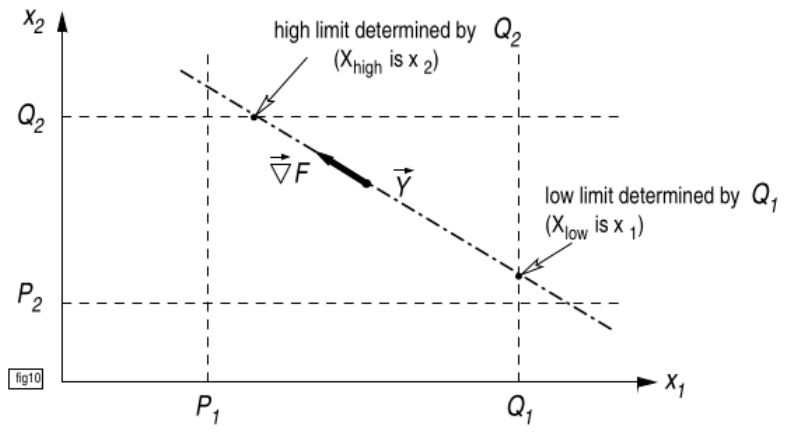


Figura 5.7: Ejemplo bidimensional del análisis del peor caso (figura 5.9 del original).

gradientes obtenidos permiten juzgar si los extremos verdaderos quedan dentro o fuera de los límites de tolerancia.

Nota de traducción: el original, al enumerar las distancias para D_{high} y D_{low}, asigna D_P y D_Q a signos del gradiente opuestos a los que acaba de deducir. Aquí se describe la selección geométrica sin repetir esa inversión. Si $\partial F/\partial X_i = 0$, F es independiente de X_i en la aproximación lineal utilizada.

5.4.5. Optimización

La optimización DC de SCEPTRE busca un mínimo local de una función objetivo respecto de variables independientes limitadas por tolerancias. Ese mínimo puede hallarse en un punto interior, con gradiente nulo, o en un límite de tolerancia. El procedimiento se basa en el método de Davidon [13]: desde una aproximación inicial calcula el gradiente y desplaza las variables, vuelve a calcularlo y repite hasta satisfacer un umbral de error. La red adjunta permite obtener el gradiente con dos cálculos de condiciones iniciales, independientemente del número de variables; además, el cálculo de la red adjunta es lineal y converge con rapidez.

Davidon combina el descenso más pronunciado con Newton–Raphson (apartado 5.4.1). Cerca de un mínimo se aproxima la función por sus términos hasta segundo orden:

$$F(X) \simeq F(Y) + h^T \nabla F(Y) + \frac{1}{2} h^T G(Y) h, \qquad h = X - Y,$$

donde G es la matriz hessiana,

$$G_{ij} = \frac{\partial^2 F}{\partial X_i \partial X_j}. \tag{5.92}$$

Al derivar la aproximación cuadrática, $\nabla F(X) \simeq \nabla F(Y) + G(Y)h$. Si el gradiente del mínimo es cero, el desplazamiento de Newton satisface $h = -G(Y)^{-1}\nabla F(Y)$. El descenso más pronunciado es simple, pero se hace lento cerca del mínimo; el método de Davidon pasa gradualmente de ese descenso al desplazamiento de Newton. En esta aproximación la inversa de la hessiana se estima iterativamente, sin invertir la matriz en cada paso.

Se comienza habitualmente con $H^{(0)} = I$, una aproximación definida positiva de G^{-1}. Si i identifica la iteración y $h^{(i)}$ el tamaño elegido por una búsqueda lineal, el original presenta el avance y la actualización como

$$
\begin{aligned}
X^{(i+1)} &= X^{(i)} + h^{(i)} H^{(i)} \nabla F^{(i)}, \\
Z &= -h^{(i)} \nabla F^{(i)}, \\
v &= \nabla F^{(i+1)} - \nabla F^{(i)}, \\
H^{(i+1)} &= H^{(i)} + A^{(i)} + B^{(i)}, \\
a_{jk}^{(i)} &= \frac{Z_j Z_k}{Z \cdot v}, \qquad b_{jk}^{(i)} = \frac{(H^{(i)} v)_j (H^{(i)} v)_k}{v \cdot H^{(i)} v}.
\end{aligned}
$$

Z y v son vectores; A y B, matrices. Según el original, A contribuye a que H converja y B conserva su carácter definido positivo. *Nota de traducción:* se conservaron los signos impresos para el avance y la actualización, aunque no concuerdan directamente con el desplazamiento de descenso $-G^{-1}\nabla F$ deducido arriba. No se deben implementar estas fórmulas sin cotejarlas con la descripción original del algoritmo y su código.

SCEPTRE modifica el procedimiento original [14, 16] para imponer los límites $P_i \le X_i \le Q_i$. Introduce una variable transformada Y_i para cada X_i:

$$
\begin{aligned}
X_i &= \frac{Q_i + P_i}{2} - \frac{Q_i - P_i}{2} \cos Y_i, \\
\frac{dX_i}{dY_i} &= \frac{Q_i - P_i}{2} \sin Y_i, \\
\frac{\partial F}{\partial Y_i} &= \frac{dX_i}{dY_i} \frac{\partial F}{\partial X_i}.
\end{aligned}
$$

Así, $X_i = P_i$ corresponde a $Y_i = 0$ y $X_i = Q_i$ a $Y_i = \pi$; la derivada dX_i/dY_i se anula en ambos límites. El gradiente respecto de las variables transformadas es, por ello, cero en los límites y puede ser no nulo en el interior. *Nota de notación:* aquí Y_i designa la nueva variable transformada; en el apartado anterior Y designaba el vector nominal.

El método original también exige que el mínimo de la función objetivo sea positivo. SCEPTRE introduce un valor de referencia variable, menor que el mínimo estimado, y lo ajusta automáticamente durante la iteración. El usuario puede fijarlo para acelerar la convergencia, pero una elección incorrecta puede empeorarla. La iteración termina cuando el cambio esperado ΔF en el próximo incremento, de $X^{(i)}$ a $X^{(i+1)}$, cae por debajo del umbral indicado; se informa entonces el valor en $X^{(i+1)}$.

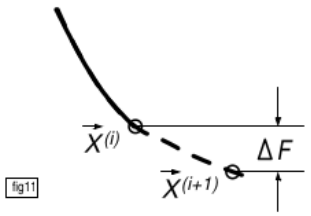


Figura 5.8: Cambio de la función objetivo entre dos iteraciones (figura 5.10 del original).

5.5. Análisis de corriente alterna (AC)

La descripción del análisis de corriente alterna distingue cuatro casos:

1. Circuitos con fuentes independientes (apartado 5.5.1): $\dot{X} = AX + GV$, con $V^T = [E_7\ J_8]$.
2. Circuitos con fuentes linealmente dependientes (apartado 5.5.2): $\dot{X} = AX + GV + Hu$, con $u^T = [E_Y\ J_X]$.
3. Circuitos que requieren derivadas temporales de las fuentes independientes (apartado 5.5.3): $\dot{X} = AX + GV + Q\dot{V}$.
4. Circuitos que reúnen los casos 2 y 3 (apartado 5.5.4): $\dot{X} = AX + GV + Q\dot{V} + Hu$.

5.5.1. Caso 1: solo fuentes independientes

Variables de estado

Un sistema lineal e invariante en el tiempo se describe, para el análisis de frecuencia, mediante

$$\dot{X} = AX + GV, \tag{5.93}$$
$$Y = CX + DV. \tag{5.94}$$

X es el vector de N variables de estado; V, el de N_S fuentes externas; e Y, el de N_R salidas solicitadas, con $N_R \leq 2N_E$ y N_E igual al número total de elementos. Las matrices reales A $(N \times N)$ y G $(N \times N_S)$ describen, respectivamente, la red y el efecto de sus fuentes; C $(N_R \times N)$ relaciona estados con salidas, y D $(N_R \times N_S)$ relaciona entradas con salidas.

Cada fuente dependiente de la frecuencia puede escribirse como

$$V_k(t, \omega) = K_k(\omega)e^{\alpha t}, \qquad 1 \leq k \leq N_S, \tag{5.95}$$

donde $\alpha = \mathrm{j}\omega$ y $K_k(\omega)$ es una función compleja de la frecuencia. En forma vectorial, y con una solución particular de la misma dependencia temporal, se obtiene

$$V = Ke^{\alpha t}, \tag{5.96}$$
$$X = X_0 e^{\alpha t}. \tag{5.97}$$

Al sustituirlas en (5.93), $\alpha X_0 = AX_0 + GK$ y, por tanto,

$$X_0 = (\alpha I - A)^{-1} GK. \tag{5.98}$$

El subprograma `EIGENP` [17, 18] diagonaliza A:

$$A = S\Lambda S^{-1}, \tag{5.99}$$

donde Λ es diagonal, de orden N, con autovalores complejos; las columnas de la matriz compleja S son los autovectores, por lo que S es la matriz modal. Al reemplazar (5.99) en (5.98), la solución AC para las variables de estado es

$$X_0 = S(\mathrm{j}\omega I - \Lambda)^{-1} S^{-1} GK(\omega). \tag{5.100}$$

La ecuación (5.94) proporciona las magnitudes que no son variables de estado. Las relaciones de capacitores e inductancias son las mismas del análisis transitorio (apartado 5.3.1), expresadas ahora en el dominio complejo:

$$I_1 = \mathrm{j}\omega C_{11} V_1, \qquad I_4 = \mathrm{j}\omega C_{44} V_4,$$
$$V_3 = \mathrm{j}\omega (L_{36} I_6 + L_{33} I_3), \qquad V_6 = \mathrm{j}\omega (L_{36}^T I_3 + L_{66} I_6).$$

Cálculo de las matrices A y G

En SCEPTRE, las corrientes I_3 de inductancias y las tensiones V_4 de capacitores forman el estado. Las fuentes independientes de tensión y de corriente forman la entrada:

$$X = \begin{bmatrix} I_3 \\ V_4 \end{bmatrix}, \qquad V = \begin{bmatrix} E_7 \\ J_8 \end{bmatrix}.$$

La ecuación (5.93) se divide en bloques:

$$\begin{bmatrix} \dot{I}_3 \\ \dot{V}_4 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} I_3 \\ V_4 \end{bmatrix} + \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} E_7 \\ J_8 \end{bmatrix}.$$

Es decir, $\dot{I}_3 = A_{11} I_3 + A_{12} V_4 + G_{11} E_7 + G_{12} J_8$ y $\dot{V}_4 = A_{21} I_3 + A_{22} V_4 + G_{21} E_7 + G_{22} J_8$.

Las ecuaciones transitorias del apartado 5.3.1 se emplean para deducir los bloques correspondientes a los elementos pasivos e invariantes en el tiempo R, L y C. Con la notación de matrices inversas del original,

$$M_{\mathrm{LI}} = [L_{33} + L_{36} B_{36}^T + B_{36} L_{63} + B_{36} L_{66} B_{36}^T]^{-1},$$
$$M_{\mathrm{RI}} = [R_{22} + B_{25} R_{55} B_{25}^T]^{-1},$$
$$M_{\mathrm{SI}} = [S_{11} + B_{14} S_{44} B_{14}^T]^{-1}.$$

Las cuatro submatrices de A son

$$A_{11} = -M_{\mathrm{LI}} B_{35} R_{55} [B_{35}^T - B_{25}^T M_{\mathrm{RI}} B_{25} R_{55} B_{35}^T], \tag{5.101}$$
$$A_{12} = -M_{\mathrm{LI}} [B_{34} - B_{35} R_{55} B_{25}^T M_{\mathrm{RI}} B_{24}], \tag{5.102}$$
$$A_{21} = C_{44I} [B_{14}^T M_{\mathrm{SI}} B_{14} S_{44} - I][B_{24}^T M_{\mathrm{RI}} B_{25} R_{55} B_{35}^T - B_{34}^T], \tag{5.103}$$
$$A_{22} = C_{44I} [B_{14}^T M_{\mathrm{SI}} B_{14} S_{44} - I][B_{24}^T M_{\mathrm{RI}} B_{24}]. \tag{5.104}$$

Para las fuentes independientes de corriente y tensión se obtienen los bloques de G:

$$G_{11} = -M_{\mathrm{LI}} [B_{37} - B_{35} R_{55} B_{25}^T M_{\mathrm{RI}} B_{27}], \tag{5.105}$$
$$G_{12} = -M_{\mathrm{LI}} B_{35} R_{55} [B_{85}^T - B_{25}^T M_{\mathrm{RI}} B_{25} R_{55} B_{85}^T], \tag{5.106}$$
$$G_{21} = C_{44I} [B_{14}^T M_{\mathrm{SI}} B_{14} S_{44} - I] B_{24}^T M_{\mathrm{RI}} B_{27}, \tag{5.107}$$
$$G_{22} = C_{44I} [B_{14}^T M_{\mathrm{SI}} B_{14} S_{44} - I][B_{24}^T M_{\mathrm{RI}} B_{25} R_{55} B_{85}^T - B_{84}^T]. \tag{5.108}$$

El símbolo C_{44I} se conserva como aparece en estas fórmulas del original. No se define de manera explícita en este apartado; antes de usar las expresiones como especificación de implementación conviene verificar su significado en la formulación inglesa [3].

5.5.2. Caso 2: circuitos con fuentes linealmente dependientes

Reducción a la forma canónica

Las fuentes de tensión y corriente que dependen linealmente de tensiones y corrientes en resistencias se designan E_Y y J_X. Si $u^T = [E_Y\ J_X]$, la ecuación de estado es

$$\dot{X} = AX + GV + Hu, \tag{5.109}$$

donde H es una matriz real de tamaño $N \times N_D$ y N_D es el número de fuentes linealmente dependientes. Se busca la forma canónica $\dot{X} = A_{\mathrm{EXT}}X + G_{\mathrm{EXT}}V$, con matrices reales de tamaños $N \times N$ y $N \times N_S$, respectivamente. En bloques, (5.109) se escribe

$$\begin{bmatrix} \dot{I}_3 \\ \dot{V}_4 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} I_3 \\ V_4 \end{bmatrix} + \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} E_7 \\ J_8 \end{bmatrix} + \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} E_Y \\ J_X \end{bmatrix}. \tag{5.110}$$

Las dependencias de esas fuentes se representan mediante matrices $k_1, \ldots, k_4$:

$$E_Y = k_1 V_2 + k_2 V_5, \tag{5.111}$$
$$J_X = k_3 I_2 + k_4 I_5. \tag{5.112}$$

Al usar $V_2 = R_{22} I_2$ e $I_5 = G_{55} V_5$, quedan agrupadas como

$$u = K \begin{bmatrix} I_2 \\ V_5 \end{bmatrix}, \qquad K = \begin{bmatrix} k_1 R_{22} & k_2 \\ k_3 & k_4 G_{55} \end{bmatrix}. \tag{5.113}$$

La sustitución de (5.113) en (5.110) da

$$\dot{X} = AX + GV + HK \begin{bmatrix} I_2 \\ V_5 \end{bmatrix}. \tag{5.114}$$

El vector de corrientes y tensiones resistivas es un subconjunto de las salidas de (5.94); por tanto,

$$\begin{bmatrix} I_2 \\ V_5 \end{bmatrix} = Y' = C'X + D'V. \tag{5.115}$$

De las ecuaciones del apartado 5.3.4 se obtiene la matriz de este subsistema:

$$M_{\mathrm{RG}} = \begin{bmatrix} R_{22} + B_{2Y} k_1 R_{22} & B_{25} + B_{2Y} k_2 \\ -B_{25}^T - B_{X5}^T k_3 & G_{55} - B_{X5}^T k_4 G_{55} \end{bmatrix}.$$

Con ella, los bloques que convierten el estado y las fuentes independientes en Y' son

$$C' = M_{\mathrm{RG}}^{-1} \begin{bmatrix} 0 & -B_{24} \\ B_{35}^T & 0 \end{bmatrix}, \tag{5.116}$$

$$D' = M_{\mathrm{RG}}^{-1} \begin{bmatrix} -B_{27} & 0 \\ 0 & B_{85}^T \end{bmatrix}. \tag{5.117}$$

En efecto, $[I_2\ V_5]^T = M_{\mathrm{RG}}^{-1}[-B_{24}V_4 - B_{27}E_7 \ \ B_{35}^T I_3 + B_{85}^T J_8]^T$. Al insertar (5.115) en (5.114), se obtiene la forma canónica buscada:

$$\dot{X} = \underbrace{(A + HKC')}_{A_{\mathrm{EXT}}} X + \underbrace{(G + HKD')}_{G_{\mathrm{EXT}}} V. \tag{5.118}$$

La notación compacta (5.118) representa exactamente las matrices por bloques desarrolladas en el original: A, G y H están en (5.110), K en (5.113), y C' y D' en (5.116) y (5.117). *Nota de traducción:* el término R_{22} en la esquina superior izquierda de M_{RG} aparece tipográficamente dañado en el PDF alemán; se transcribe según la relación $V_2 = R_{22} I_2$ y la estructura de (5.113).

Cálculo de la matriz H

La construcción de H es análoga a la de G. La concatenación de ambas multiplica al vector de fuentes $[E_7\ J_8\ E_Y\ J_X]^T$; por eso sus bloques difieren solamente en las submatrices de topología que corresponden a cada clase de fuente. El original los presenta como

$$G = \begin{bmatrix} g_1 B_{37} + g_2 B_{27} & g_4 B_{85}^T \\ g_3 B_{27} & g_5 B_{85}^T + g_6 B_{84}^T \end{bmatrix},$$
$$H = \begin{bmatrix} g_1 B_{3Y} + g_2 B_{2Y} & g_4 B_{X5}^T \\ g_3 B_{2Y} & g_5 B_{X5}^T + g_6 B_{X4}^T \end{bmatrix}.$$

Los coeficientes matriciales comunes son

$$\begin{aligned} g_1 &= -M_{\mathrm{LI}}, \\ g_2 &= M_{\mathrm{LI}}(B_{35} R_{55})(B_{25}^T M_{\mathrm{RI}}), \\ g_3 &= C_{44I}(B_{14}^T M_{\mathrm{SI}} B_{14} S_{44} - I) B_{24}^T M_{\mathrm{RI}}, \\ g_4 &= -M_{\mathrm{LI}} B_{35} R_{55}[I - (B_{25}^T M_{\mathrm{RI}})(B_{25} R_{55})], \\ g_5 &= C_{44I}(B_{14}^T M_{\mathrm{SI}} B_{14} S_{44} - I) B_{24}^T M_{\mathrm{RI}} B_{25} R_{55}, \\ g_6 &= -C_{44I}(B_{14}^T M_{\mathrm{SI}} B_{14} S_{44} - I). \end{aligned}$$

5.5.3. Caso 3: circuitos que requieren derivadas temporales de fuentes independientes

Deducción de la solución AC

Algunos circuitos requieren, para el análisis transitorio, la derivada temporal de una fuente: sucede, por ejemplo, en mallas de fuentes independientes de tensión y capacitores, o en nodos con fuentes independientes de corriente e inductancias. El análisis AC necesita las mismas derivadas, pero SCEPTRE las calcula y el usuario no tiene que proporcionarlas. El sistema se escribe

$$\dot{X} = AX + GV + Q\dot{V}, \qquad V_k(t, \omega) = K_k(\omega) e^{\mathrm{j}\omega t}. \tag{5.119}$$

Con $X = X_0 e^{\mathrm{j}\omega t}$, la sustitución en (5.119) da

$$(\mathrm{j}\omega I - A) X_0 = GK + \mathrm{j}\omega QK, \qquad X_0 = (\mathrm{j}\omega I - A)^{-1}(GK + \mathrm{j}\omega QK).$$

Al utilizar la transformación de semejanza $A = S\Lambda S^{-1}$ de (5.99), se obtiene

$$X_0 = S(\mathrm{j}\omega I - \Lambda)^{-1}[S^{-1} GK + \mathrm{j}\omega S^{-1} QK]. \tag{5.120}$$

El segundo término entre corchetes incorpora las derivadas de las fuentes. *Nota de traducción:* en (5.120) el PDF alemán imprime A dentro de la inversa, aunque después de la transformación de semejanza corresponde la matriz diagonal Λ; aquí se escribe Λ para que concuerde con (5.99) y (5.100).

Cálculo de la matriz Q

La matriz se divide en bloques $Q = \left[\begin{smallmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{smallmatrix}\right]$, con

$$
\begin{aligned}
Q_{11} &= 0, \\
Q_{12} &= -M_{\mathrm{LI}}[B_{36} L_{66} B_{86}^T + L_{36} B_{86}^T], \\
Q_{21} &= -C_{44I} B_{14}^T M_{\mathrm{SI}} B_{17}, \\
Q_{22} &= 0.
\end{aligned}
$$

5.5.4. Caso 4: fuentes linealmente dependientes y derivadas de fuentes independientes

Este caso combina los procedimientos de los apartados 5.5.2 y 5.5.3: $\dot{X} = AX + GV + Q\dot{V} + Hu$. El manual alemán no desarrolla aquí otras ecuaciones.

A Flujo del programa

SCEPTRE se escribió originalmente en FORTRAN IV para el IBM 7090/94 y el IBM System/360 (OS/360). El volumen II del manual original [3] explica cada etapa del programa; la presente obra alemana omite esas explicaciones. Para el funcionamiento de la versión revisada SC1/SD2 bajo Linux/UNIX, remite al manual del usuario [4].

Los siete diagramas siguientes muestran el flujo de la versión para IBM 7090/94 y OS/360. Se reproducen sin modificar sus rótulos originales en inglés; no describen necesariamente la organización interna de SC1/SD2.

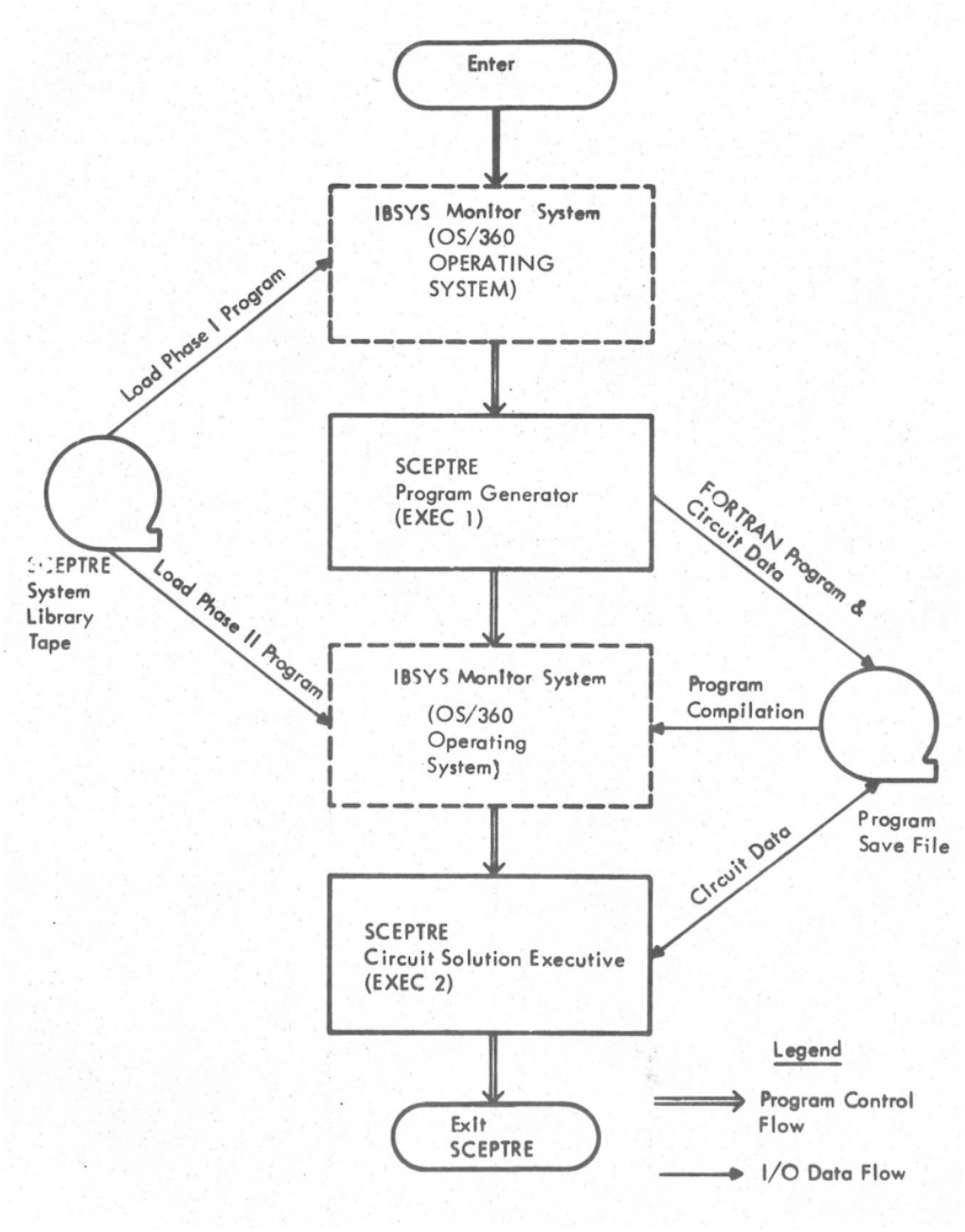


Figura A.1: Flujo general del sistema (figura A.1 del original).

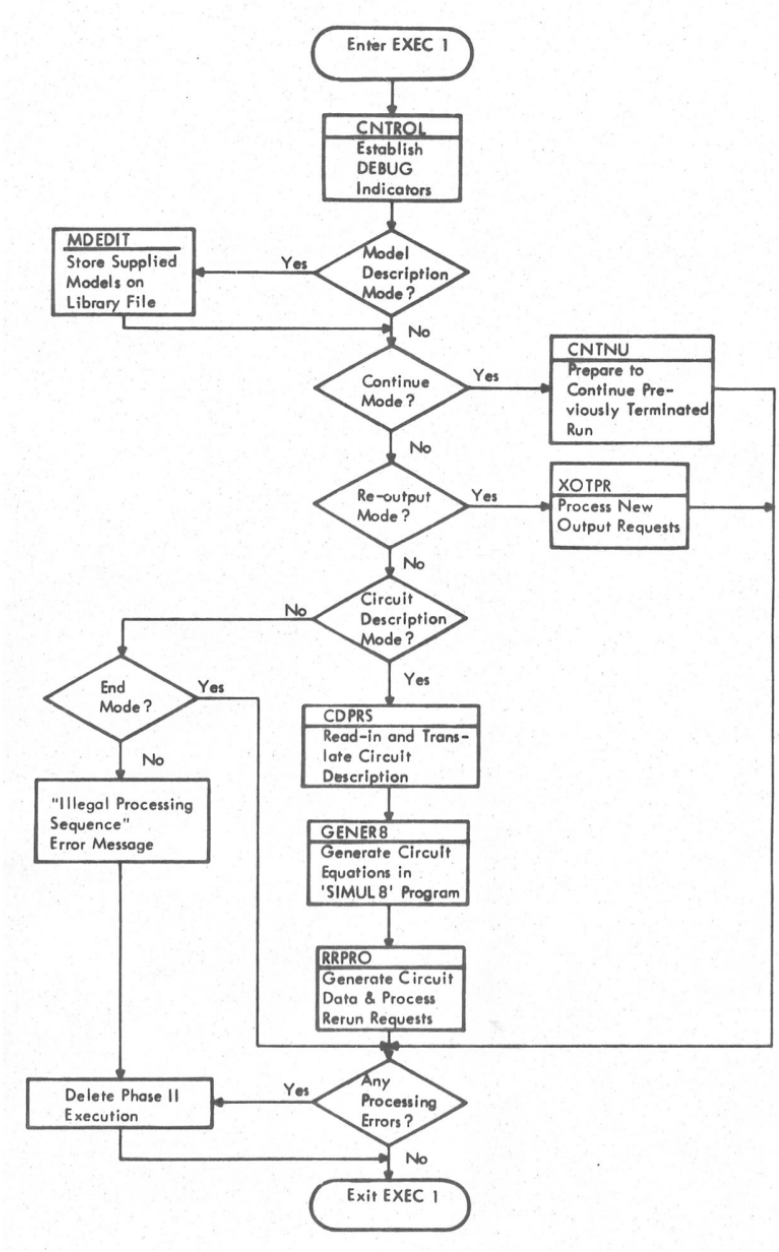


Figura A.2: Generador del programa, SCEPTREM (figura A.2 del original).

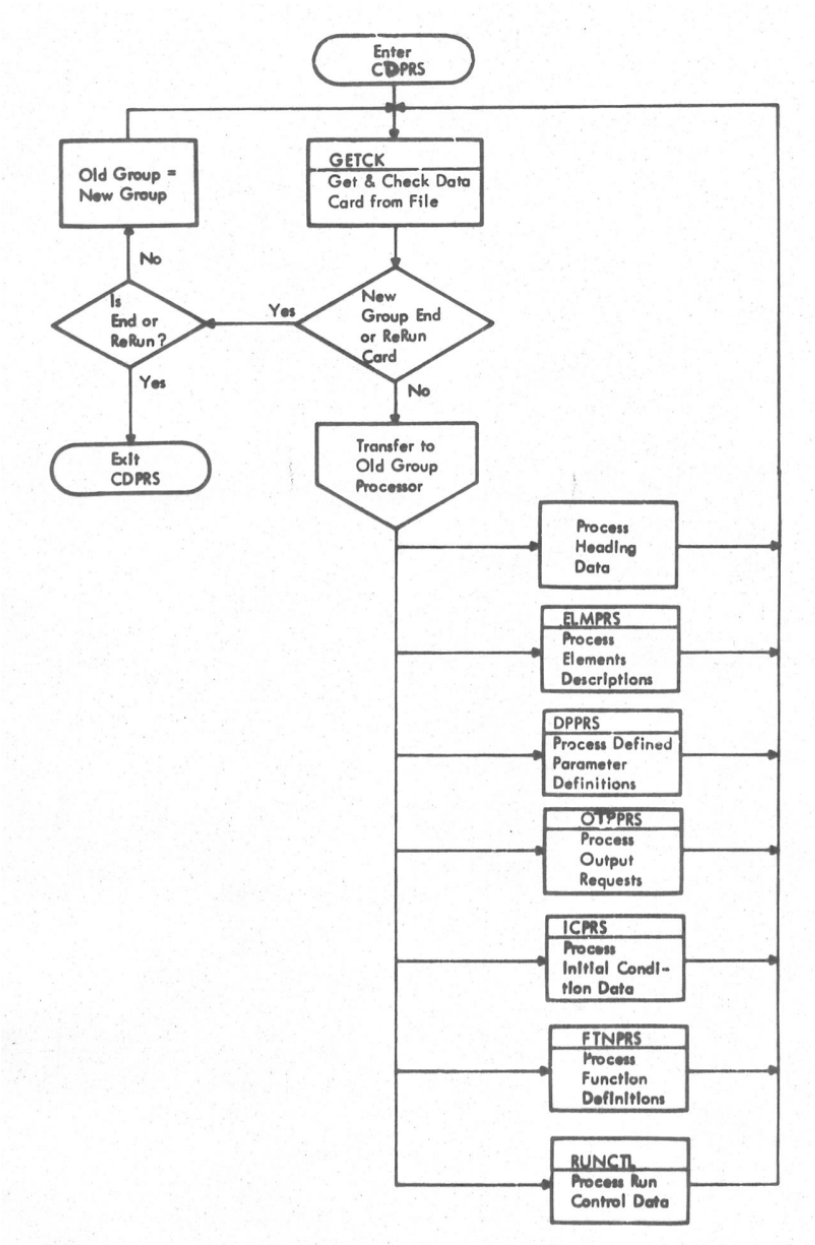


Figura A.3: Procesador de la descripción del circuito, CDPRS (figura A.3 del original).

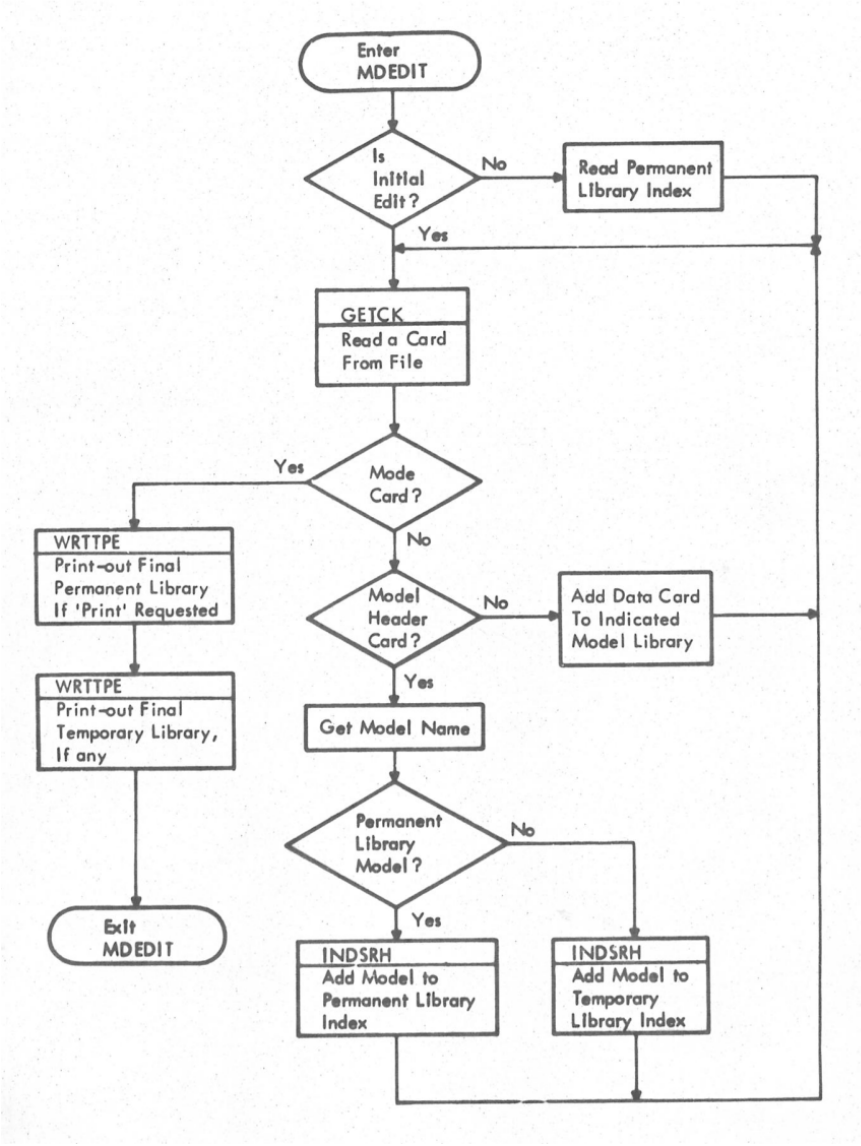


Figura A.4: Editor de modelos, MDEDIT (figura A.4 del original).

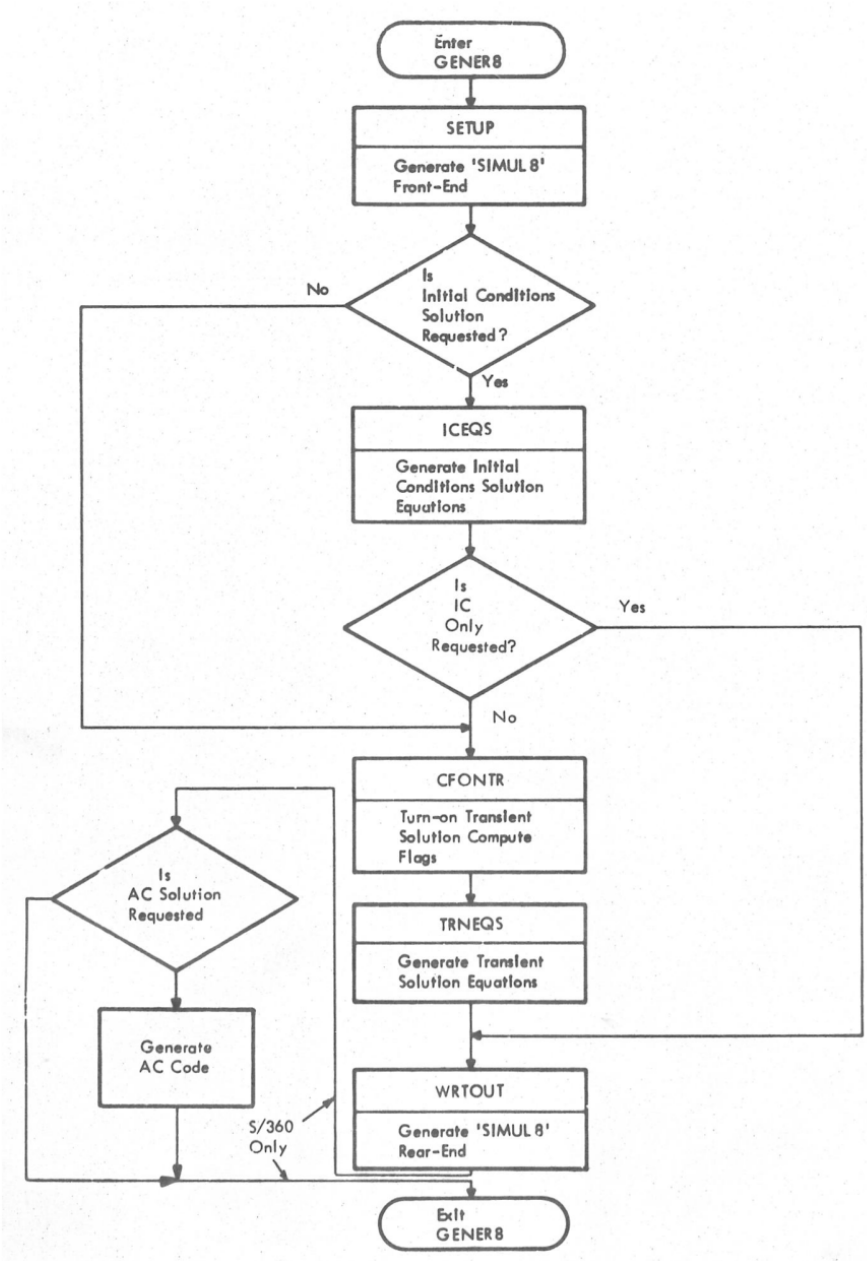


Figura A.5: Generador de ecuaciones del circuito, GENER8 (figura A.5 del original).

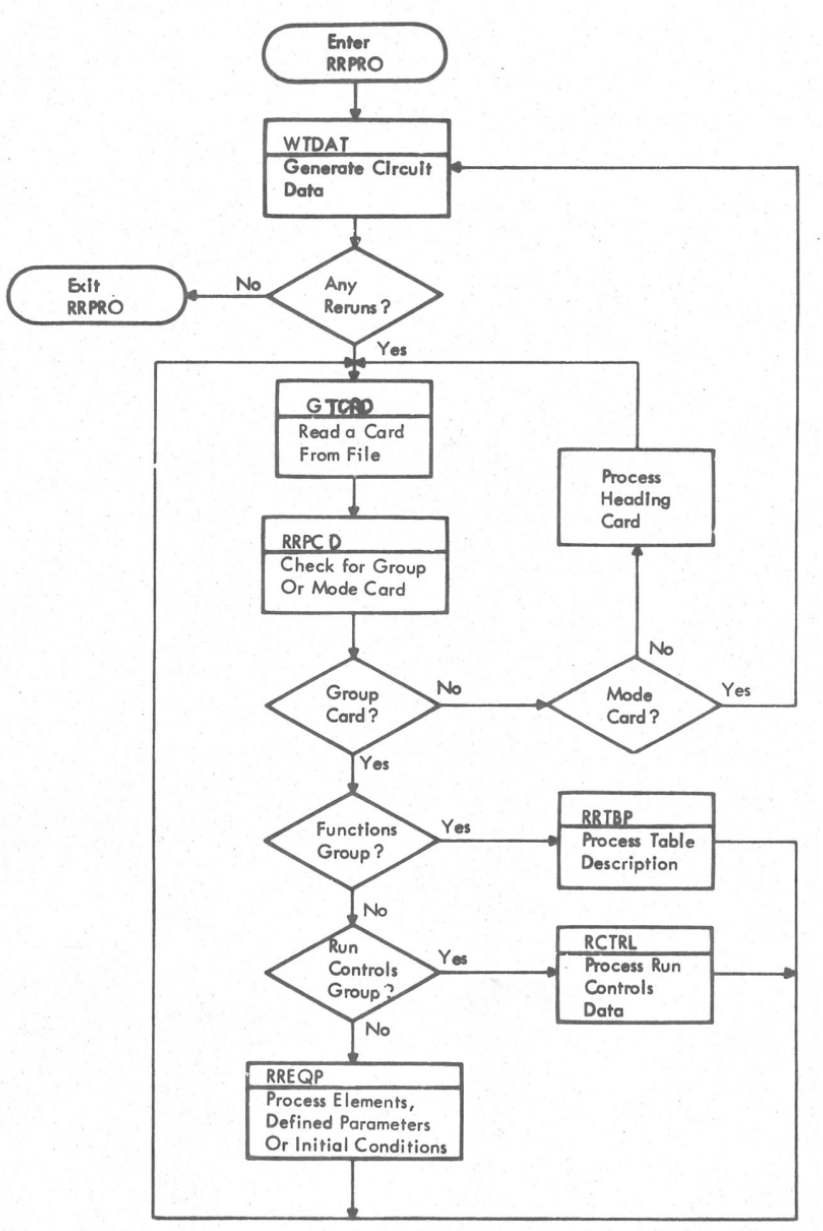


Figura A.6: Procesador de nuevas corridas y generador de datos, RRPRO (figura A.6 del original).

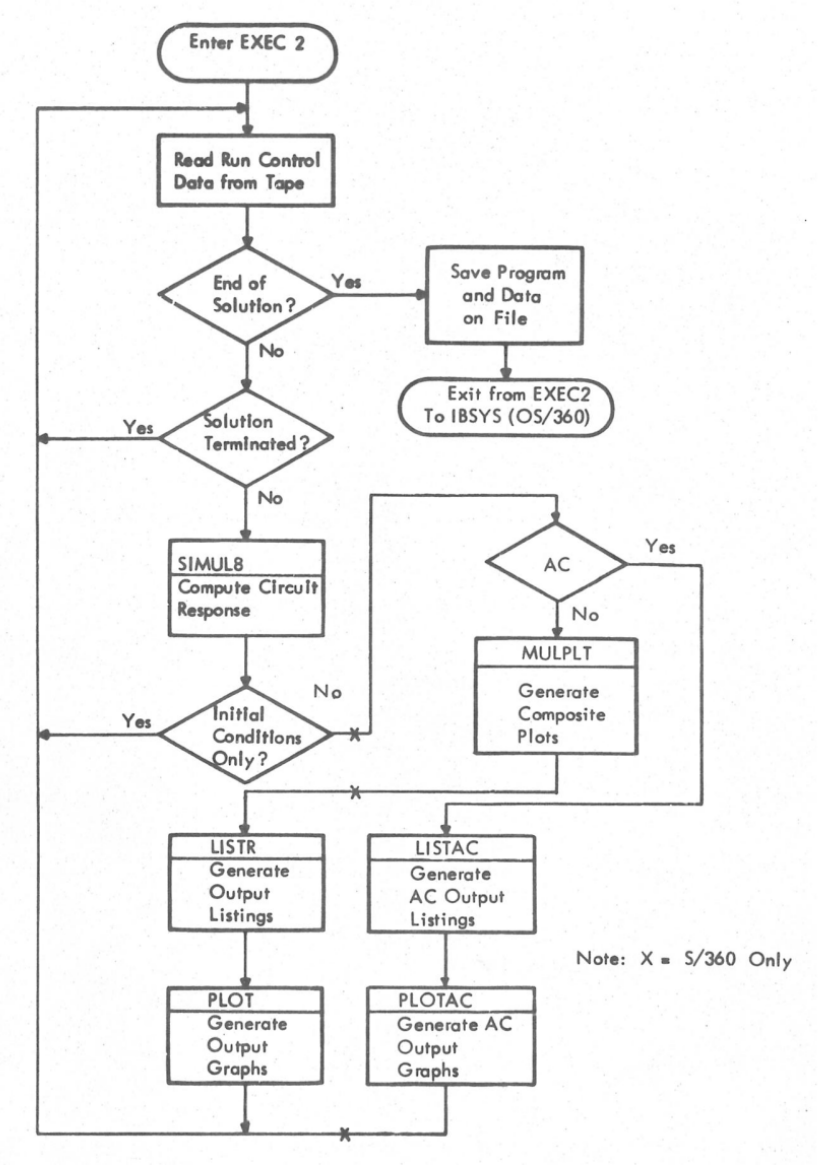
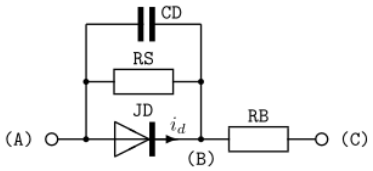


Figura A.7: Ejecutivo de solución del circuito, SCEPTREDU (figura A.7 del original).

B Circuitos equivalentes de diodos y transistores

B.1. Diodo

Existen distintos modelos para simular dispositivos semiconductores. Las figuras B.1, B.2 y B.3 del original muestran los circuitos equivalentes de un diodo y de transistores bipolares pnp y npn, junto con su representación en SCEPTRE. Se basan en el modelo de Ebers–Moll. Los parámetros indicados en cursiva en las instrucciones de SCEPTRE deben obtenerse de las hojas de datos de los semiconductores correspondientes, manteniendo un sistema coherente de unidades.

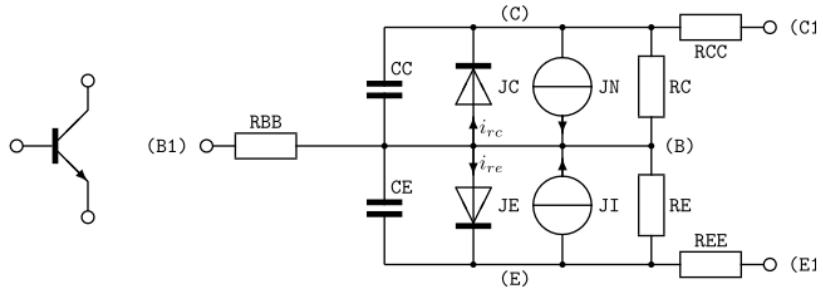


Parameter: $i_s, \Theta, R_B, R_S, C_0, V_0, n, K$	MODEL *name* (A-C) ELEMENTS
$i_d = i_s(e^{\Theta V_{\mathrm{JD}}} - 1)$ $C_D = \dfrac{C_0}{(V_0 - V_{\mathrm{JD}})^n} + K(i_d + i_s)$	JD, A-B = DIODE EQUATION (i_s, Θ) CD, A-B = Q1 ($C_0, V_0, V_{\mathrm{JD}}, n, K, J_D, i_s$) RB, B-C = R_B RS, A-B = R_S FUNCTIONS Q1 (A,B,C,D,E,F,G)=(A/(B-C)**D+E*(F+G))

Figura B.1: Circuito equivalente de un diodo y su modelo SCEPTRE (figura B.1 del original).

B.2. Transistor pnp

El esquema, las ecuaciones del modelo y sus instrucciones SCEPTRE se reproducen en la figura B.2.

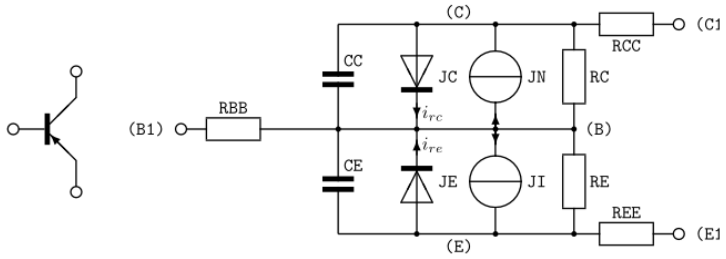


Parameter: $i_{es}, \Theta_e, \alpha_i, i_{cs}, \Theta_c, \alpha_n, C_{0e}, V_{0e}, K_e, n_e, C_{0c}, V_{0c}, K_c, n_c, R_{BB}, R_{EE}, zR_{CC}, R_E, R_C$	`MODEL` *name* `(B1-E1-C1)`
	`ELEMENTS`
$i_{re} = i_{es}(e^{\Theta_e V_{BE}} - 1)$	`JE, B-E =` `DIODE EQUATION` (i_{es}, Θ_e)
$i_{rc} = i_{cs}(e^{\Theta_c V_{BC}} - 1)$	`JC, B-C =` `DIODE EQUATION` (i_{cs}, Θ_c)
$i_i = \alpha_i i_{rc}$	`JI, E-B =` α_i `* JC`
$i_n = \alpha_n i_{re}$	`JN, C-B =` α_n `* JE`
$C_e = \frac{C_{0e}}{(V_{0e} - V_{BE})^{n_e}} + K_e(i_{re} + i_{es})$	`CE, B-E = Q1` $(C_{0e}, V_{0e}, V_{CE}, n_e, K_e, J_E, i_{es})$
$C_c = \frac{C_{0c}}{(V_{0c} - V_{BC})^{n_c}} + K_c(i_{rc} + i_{cs})$	`CC, B-C = Q1` $(C_{0c}, V_{0c}, V_{CC}, n_c, K_c, J_C, i_{cs})$
	`RBB, B1-B =` R_{BB}
	`REE, E1-E =` R_{EE}
	`RCC, C1-C =` R_{CC}
	`RE, E-B =` R_E
	`RC, C-B =` R_C
	`FUNCTIONS`
	`Q1 (A,B,C,D,E,F,G)=(A/(B-C)**D+E*(F+G))`

Figura B.2: Transistor pnp según el título del apartado B.2 del original.

B.3. Transistor npn

La figura B.3 presenta el circuito equivalente de polaridad opuesta y la descripción SCEPTRE correspondiente.



Parameter: $i_{es}, \Theta_e, \alpha_i, i_{cs}, \Theta_c, \alpha_n,$ $C_{0e}, V_{0e}, K_e, n_e, C_{0c}, V_{0c}, K_c, n_c,$ $R_{BB}, R_{EE}, R_{CC}, R_E, R_C$	MODEL *name* (B1-E1-C1) ELEMENTS
$i_{re} = i_{es}(e^{\Theta_e V_{BE}} - 1)$	JE, E-B = DIODE EQUATION (i_{es}, Θ_e)
$i_{rc} = i_{cs}(e^{\Theta_c V_{BC}} - 1)$	JC, C-B = DIODE EQUATION (i_{cs}, Θ_c)
$i_i = \alpha_i i_{rc}$	JI, B-E = α_i * JC
$i_n = \alpha_n i_{re}$	JN, B-C = α_n * JE
$C_e = \dfrac{C_{0e}}{(V_{0e} - V_{BE})^{n_e}} + K_e(i_{re} + i_{es})$	CE, E-B = Q1 ($C_{0e}, V_{0e}, V_{CE}, n_e, K_e, J_E, i_{es}$)
$C_c = \dfrac{C_{0c}}{(V_{0c} - V_{BC})^{n_c}} + K_c(i_{rc} + i_{cs})$	CC, C-B = Q1 ($C_{0c}, V_{0c}, V_{CC}, n_c, K_c, J_C, i_{cs}$)
	RBB, B1-B = R_{BB}
	REE, E1-E = R_{EE}
	RCC, C1-C = R_{CC}
	RE, E-B = R_E
	RC, C-B = R_C
	FUNCTIONS
	Q1 (A,B,C,D,E,F,G)=(A/(B-C)**D+E*(F+G))

Figura B.3: Transistor npn según el título del apartado B.3 del original.

Nota de traducción: en el PDF alemán, los pies de las figuras B.2 y B.3 intercambian «npn» y «pnp» respecto de los títulos de sus apartados. Aquí se sigue la denominación de los títulos y se conservan sin cambios los esquemas y las instrucciones impresas dentro de las figuras.

C Apéndices del volumen I [2]

C.1. Resumen de las instrucciones de SCEPTRE

La tabla C.1 relaciona las instrucciones de SCEPTRE con las variables internas correspondientes del bloque `COMMON /CNTRLS/`. El índice se refiere al vector `CTROLS` del mismo bloque, descrito en el apartado C.2. Las celdas vacías reproducen las ausencias del original; no se les asigna una variable por inferencia.

Tabla C.1: Instrucciones de SCEPTRE y variables del bloque `CNTRLS` (tabla C.1 del original).

Instrucción SCEPTRE	**Variable COMMON**	**Índice CTROLS**
START TIME	TIME	1
STOP TIME	XSTOPT	2
MAXIMUM STEP SIZE	XMXISS	7
INTEGRATION ROUTINE	XIR	4
STARTING STEP SIZE	XTISSS	5
MINIMUM STEP SIZE	XMNISS	6
MINIMUM ABSOLUTE ERROR	XMNAIE	8
MAXIMUM ABSOLUTE ERROR	XMXAIE	9
RUN INITIAL CONDITIONS	XIC	10
COMPUTER TIME LIMIT	XMXERT	3
RUN INITIAL CONDITIONS ONLY	XTRANS	34
MINIMUM RELATIVE ERROR	XMNRIE	13
MAXIMUM RELATIVE ERROR	XMXRIE	14
RELATIVE CONVERGENCE	XICRER	17
ABSOLUTE CONVERGENCE	XICAER	18
X PLOT DIMENSION	XXDIMS	20
Y PLOT DIMENSION	XYDIMS	21
MAXIMUM PRINT POINTS	XMXOPT	19
NO ELEMENT SORT		
MAXIMUM INTEGRATION PASSES	XMXPAS	12
NEWTON-RAPHSON PASS LIMIT	XMXICP	16
PUNCH BINARY CARDS		
FULIST		
PUNCH PROGRAM		
WRITE SIMUL8 DATA		
COMPUTER SAVE INTERVAL	XSAVE	35
VECTOR EQUATIONS		
WRITE DEBUG		
PRINT B MATRIX		
SOLUTION TIME LIMIT	XMXSLT	37
PLOT INTERVAL	XPLTI	38
PRINT INTERVAL	XPRTI	39
LIST NODE MAP		
IC FOR RERUNS	XSAVIC	53
RUN IC VIA IMPLICIT	XTRNIC	55
USE DIFFERENCED JACOBIAN	XDIFF	61
USE SYMBOLIC JACOBIAN	XSYMBL	62
EXECUTE SETUP PHASE ONLY		

Continuación de la tabla C.1

Instrucción SCEPTRE	Variable COMMON	Índice CTROLS
RUN AC	XAC	71
INITIAL FREQUENCY	XINFRQ	72
FINAL FREQUENCY	XFNFRQ	73
TYPE FREQUENCY	XTYPFQ	74
NUMBER FREQUENCY STEPS	XNFRQS	75
PRINT A MATRIX		
PRINT EIGENVALUES		
PRINT EIGENVECTORS		
RUN MONTE CARLO	XNMCPS	121
RUN OPTIMIZATION	XNOPPS	131
RUN WORST CASE	XWCLHN	151
FREQUENCY	FREQ	79
INITIAL RANDOM NUMBER	XRANNO	122
DISTRIBUTION	XDISTR	125
LIST MONTE CARLO DETAILS		
USE FIXED AC MATRIX IN RERUNS	XACRE	85
IMPULSE RESPONSE BUFFER	XHAVE	87
INPUT FUNCTION BUFFER	XIAVE	88
COMPRESSION CRITERION	XCMPCR	89
COMPRESSION COUNT	XCMPCT	90
LIST OPTIMIZATION DETAILS		
PUNCH OPTIMIZATION RESULTS		
OPTIMIZATION CRITERION	XOCPR	134
OPTIMIZATION RANDOM STEPS	XNOPRS	135
RANDOM STEP SIZE CONTROL	XRNSSC	136
INITIAL H MATRIX FACTOR	XHMFAC	137
RUN SENSITIVITY	XSENST	141
MINIMUM FUNCTION ESTIMATE	XMNFES	138

C.2. Variables internas: bloque COMMON /CNTRLS/

Se puede acceder a las variables internas por su nombre o mediante el vector CTROLS y el índice de la tabla C.1. El original reproduce la siguiente declaración FORTRAN:

```
 IMPLICIT REAL*8(A-J,L-M,O-Z),INTEGER*4(K,N)
 COMMON /CNTRLS/
*   TIME ,XSTOPT,XMXERT,XIR    ,XTISSS,XMNISS,XMXISS,
*   XMNAIE,XMXAIE,XIC   ,XRERNO,XMXPAS,XMNRIE,XMXRIE,
*   XMNISO,XMXICP,XICRER,XICAER,XMXOTP,XXDIMS,XYDIMS,
*   XICPAS,XNOPRQ,XNTNDX,XNHEAD,XNDFEQ,XERT ,XNRERN,
*   XCNTNU,XSTPNO,XPASNO,XRUNNO,XSTPSZ,XTRANS,XSAVE ,
*   XTMON ,XMXSLT,XPLTI ,XPRTI ,XMCPUT,SP1(3),XWRTSM,
*   SP2(8),XSAVIC,XGER ,XTRNIC,SP3(5),XDIFF ,XSYMBL,
*   SP4(8),XAC   ,XINFRQ,XFNFRQ,XTYPFQ,XNFRQS,SP5(3),
*   FREQ ,XNWDBM,SP6(3),XDERIV,XACRE ,XNCONV,XHAVE ,
*   XIAVE ,XCMPCR,XCMPCT,XNEY ,XNE7DT,XNBC ,XNBR ,
*   XNBL ,XNLC ,XNLR ,XNLL ,XNJX ,XNJ8DT,XNJ0 ,
*   XNJ9 ,XNJ8AC,XNE7AC,XNM33 ,XNM36 ,XNM66 ,XNDP ,
*   XNDPD ,XNELTS,XLSW ,XLSWP ,SP7(2),XICPRT,SP8(5),
*   XNMCPS,XRANNO(2),SP9,XDISTR,SPA(5),XNOPPS,SPB(2),
*   XOPCR ,XNOPRS,XRNSSC,XHMFAC,XMNFES,SPC(2),XSENST,
*   XADJNT,SPD(8),XWCLHN,SPE(9)
 DIMENSION CTROLS(160)
 EQUIVALENCE (CTROLS,TIME)
```

El manual advierte expresamente que no deben modificarse los valores de las variables de control. *Nota de traducción:* la tabla C.1 escribe `XMXOPT` y `XOCPR`, mientras que la declaración imprime `XMXOTP` y `XOPCR`, respectivamente; se conservan ambas grafías para que pueda cotejarse la discrepancia con el código fuente.

C.3. Capacidad del programa

La tabla C.2 recoge los límites de SCEPTRE indicados por el original. Si se excede uno, la ejecución se interrumpe con un mensaje de error. Son límites de la versión documentada allí; no deben atribuirse sin comprobación a otra implementación.

Tabla C.2: Capacidad del programa (tabla C.2 del original).

Recurso	**Máximo**
Elementos de la red	300
Nodos	301
Derivadas de fuentes DE, DJ	50
Parámetros definidos P	200
Parámetros definidos W	50
Parámetros definidos DP	200
Inductancias mutuas M	50
Argumentos de instrucciones EQUATION	50
Cambios de tablas en modelos	15
Cambios de EQUATION en modelos	15
Supresiones de salida en modelos	10
Variables de salida	100
Condiciones iniciales	100
Instrucciones EQUATION	unos 5000 caracteres en total
Líneas de continuación de una instrucción EQUATION	20
Tablas	800 pares de valores en total
Condiciones de terminación TERMINATE, STOP	10
Modelos guardados permanentemente	250
Nodos externos de modelos	25
Nodos internos de modelos	301
Fuentes AC de primer y segundo orden	50
Fuentes independientes AC de corriente y tensión	50
Tarjetas de comentario	11

D Apéndices del volumen II [3]

D.1. Matrices dispersas

Para resolver un sistema lineal

$$AX = b, \tag{D.1}$$

con A como matriz de coeficientes de orden n, b como vector de excitaciones y X como vector de incógnitas, puede emplearse la eliminación de Gauss. El original señala que el costo computacional aumenta considerablemente con n y que la matriz contiene muchos ceros. Para redes grandes ($n > 100$) propone la factorización LU; una nota del autor aclara que su apreciación del costo data de 1974.

Si existe A^{-1}, se factoriza la matriz en una triangular inferior L y una triangular superior U de diagonal unitaria:

$$A = LU. \tag{D.2}$$

Los coeficientes se obtienen recursivamente, para $i \geq q$ y $p < j$, mediante

$$l_{iq} = a_{iq} - \sum_{k=1}^{q-1} l_{ik} u_{kq},$$
$$u_{pj} = \frac{a_{pj} - \sum_{k=1}^{p-1} l_{pk} u_{kj}}{l_{pp}},$$

con $l_{iq} = 0$ si $i < q$, $u_{pj} = 0$ si $p > j$ y $u_{pp} = 1$. La sustitución de (D.2) en (D.1) y la introducción de un vector intermedio Y dan

$$LUX = b, \tag{D.3}$$
$$Y = UX, \tag{D.4}$$
$$LY = b. \tag{D.5}$$

Primero se resuelve $LY = b$ por sustitución hacia adelante y luego $UX = Y$ por sustitución hacia atrás. Según el manual, SCEPTRE activa esta técnica solo cuando la red tiene más de 70 elementos; en redes menores usa Gauss–Jordan. Véase también [7, cap. 3.3].

D.2. Rutinas ensamblador para entrada y salida

Para ahorrar tiempo de cálculo, algunas operaciones FORTRAN de entrada y salida —`WRITE`, `READ` y `ENDFILE`— se sustituyeron por rutinas ensamblador llamadas `GET`, `PUT` y `ENDFIL`. Esto redujo la portabilidad de SCEPTRE; las instrucciones FORTRAN correspondientes quedaron comentadas en el fuente. También se usaron las rutinas `CPY`, `DAND`, `DOR`, `DEXOR`, `LBITS`, `XIBCD` y `SORT` para manipular bits y bytes, e `ITODAY` para la fecha. La versión descrita por el autor reemplazó todas esas rutinas ensamblador por rutinas FORTRAN 77.

La tabla D.1 compara tiempos de la fase I en un IBM S/360 modelo 65:

Circuito de prueba	FORTRAN I/O (s)	Ensamblador I/O (s)
Ejemplo 1, apartado 4.1	11	5
Ejemplo 2, apartado 4.2	24	9
Ejemplo 3, apartado 4.3	23	6
Ejemplo 4, apartado 4.4	11	5
Circuito de 221 elementos (solo análisis TR)	90	62

Tabla D.1: Comparación de tiempos de la fase I (tabla D.1 del original).

D.3. Método de integración implícita

La descripción detallada, en inglés, figura en el volumen II del manual original [3, anexo A].

D.4. Deducción de la matriz B

La matriz B expresa las tensiones de las ramas de enlace mediante las tensiones de las ramas del árbol, y las corrientes de las ramas del árbol mediante las corrientes de las ramas de enlace. El desarrollo detallado está en el volumen II del manual original [3, anexo B].

D.5. Evaluación de la característica del diodo al calcular condiciones iniciales

Véase el volumen II del manual original [3, anexo C].

D.6. Método de Newton–Raphson

Véase el volumen II del manual original [3, anexo D].

D.7. Identidad de las ecuaciones (5.79) y (5.80)

Véase el volumen II del manual original [3, anexo E].

D.8. Red adjunta

El cálculo de sensibilidad de un circuito emplea la red adjunta. Los detalles están en el volumen II del manual original [3, anexo F].

E Más explicaciones del SCEPTRE de Novender

Este anexo reúne explicaciones complementarias del lenguaje, los modelos, las funciones auxiliares y los formatos de resultados del SCEPTRE recibido de Wolf-Rainer Novender. Es documentación elaborada para esta edición en español, no una traducción literal de un anexo del original. Los cambios posteriores del proyecto se documentan por separado.

Los listados breves se conservan en archivos de `masexplicaciones/listados/` y se incorporan con `\lstinputlisting`. Los fragmentos de sintaxis o de modelos llevan extensión `.txt`; no son entradas completas ejecutables. El ejemplo `d01.d` se incluye completo y conserva su extensión de entrada SCEPTRE. La procedencia de cada ejemplo se indica en su sección. Las observaciones de ejecución se distinguen de las propiedades del lenguaje y no acreditan nuevas pruebas de simulación.

E.1. Elementos K y representación por convolución

Origen: función del SCEPTRE de Novender. El analizador del árbol de partida ya reconoce `FCONVE` y `FCONVJ`. Se explica aquí su representación y su sintaxis, sin incorporar las reparaciones posteriores del proyecto.

Un elemento de convolución representa un subsistema lineal mediante su respuesta impulsional, en lugar de describir cada uno de sus componentes. Para una impedancia, la relación tiene la forma

$$v(t) = \int_0^t h(t-\tau) i(\tau)\, \mathrm{d}\tau,$$

con el tratamiento de los valores iniciales y del término instantáneo que requiera el modelo. La variante de admitancia intercambia las funciones de tensión y corriente.

Fragmento de sintaxis de `a06b.d`:

```
ELEMENTS
  KCV1, 7-8 = FCONVE(1001.)
  KCV2, 8-0 = FCONVE(1002.)
  KCV3, 7-0 = FCONVE(1003.)
DEFINED PARAMETERS
  PVOLT1 = Q2(VRKCV1,EKCV1)
FUNCTIONS
  Q2(AA,BB) = (AA+BB)
OUTPUTS
  IRKCV1, PVOLT1
```

Los argumentos `1001.`, `1002.` y `1003.` identifican núcleos de respuesta impulsional; no son valores de resistencia. `FCONVE` selecciona una representación de impedancia y `FCONVJ`, una de admitancia.

En el equivalente de impedancia, la preparación introduce una resistencia y una fuente de historia. En el ejemplo, `IRKCV1` es la corriente de la resistencia equivalente y `PVOLT1` reconstruye la tensión de la rama sumando `VRKCV1` y `EKCV1`. No debe atribuirse por separado a uno de esos términos toda la tensión del subsistema.

`a06a.d` describe las cajas RLC explícitas y sirve como referencia física. El archivo de núcleos es una entrada auxiliar indispensable, explicada en la sección siguiente. La elección

de esta representación no garantiza estabilidad para cualquier paso de integración.
Fuentes: este agregado; `STCONV`, `CVPREP`, `CNVCOM`, `RES` y `CVYCLC`.

E.2. Archivo de respuestas impulsionales de la convolución

Origen: formato del SCEPTRE de Novender. Los núcleos se leen mediante la unidad Fortran 12. El nombre físico `fort.12` corresponde a la asociación de esa unidad; no es una instrucción del lenguaje `.d`.

Por cada núcleo, `RDCONV` espera estos registros:

1. Identificador entero, formato `I10`.
2. Valor de la respuesta impulsional en 0^+, un campo de 20 caracteres.
3. Valor de la función de transferencia en frecuencia cero, otro campo de 20 caracteres.
4. Número de pares tabulados, formato `I10`.
5. Pares tiempo–respuesta, dos campos numéricos de 20 caracteres por registro.

La tabla se guarda con tiempos negativos, desde $-T_{\text{máx}}$ hasta cero, como $h(-\tau)$. El último tiempo debe ser cero. Se concatenan tantos bloques como núcleos necesite el caso. Los identificadores deben coincidir con los argumentos de las llamadas `FCONVE` o `FCONVJ`.

Para las cajas RLC paralelas del par heredado `a06a.d`/`a06b.d`, en régimen subamortiguado, se emplea

$$h(t) = \frac{e^{-at}}{C}\left[\cos(\omega_d t) - \frac{a}{\omega_d}\sin(\omega_d t)\right], \quad a = \frac{1}{2RC}, \quad \omega_d = \sqrt{\frac{1}{LC} - a^2}.$$

El dato en frecuencia cero es $Z(0) = 0$ para ese paralelo RLC, debido a la inductancia ideal; no debe confundirse con una resistencia arbitraria de entrada.

Los campos numéricos tienen ancho fijo. En el lector de partida se usa `G20.8`; para evitar una escala implícita no deseada, los números deben llevar punto decimal o exponente explícito. No basta escribir números separados por espacios sin respetar los anchos.
Fuentes: `sceptre2novender/nuevo/sceptreDU/rdconv.f90`, este agregado y `sceptre2novender/corpus/fort.12`.

E.3. Dimensionamiento de los búferes de convolución

Origen: controles del SCEPTRE de Novender. Ambos controles aparecen en el árbol de partida; el manual sólo los enumera.

```
RUN CONTROLS
  IMPULSE RESPONSE BUFFER = 300
  INPUT FUNCTION BUFFER = 1200
```

`IMPULSE RESPONSE BUFFER` reserva espacio para los datos de respuesta impulsional que utiliza la preparación de los elementos de convolución. `INPUT FUNCTION BUFFER` dimensiona el historial de entrada por elemento que usa el cálculo. No representan una cantidad de variables solicitadas en `OUTPUTS`.

Sin compresión, un historial de duración T con paso aproximado Δt necesita del orden de

$$N \simeq 1 + \frac{T}{\Delta t},$$

además de posiciones para la inicialización y las evaluaciones del algoritmo. Es una estimación de dimensionamiento, no una cota exacta del programa.

La preparación y la ejecución usan estructuras diferentes: no debe trasladarse automáticamente al primer control la estimación anterior del historial temporal. Hay que comprobar también cuántos pares contienen los núcleos efectivamente cargados.

La memoria necesaria aumenta con la cantidad de elementos de convolución y con la longitud de sus historiales. La compresión puede reducir los puntos retenidos, pero no autoriza a suponer que un búfer pequeño siempre alcanzará. Si la tabla se llena, `CVYCLC` señala `CONVOLUTION TABLE HAS BECOME FULL AT TIME=...`; ese resultado no equivale a completar normalmente la simulación.
Fuentes: `CVPREP`, `CVYCLC` y este agregado.

E.4. Compresión del historial de convolución

Origen: mecanismo del SCEPTRE de Novender. La compresión actúa sobre el historial de entrada de la convolución: no comprime el archivo Dirac ni la tabla impresa de resultados.

Ejemplo de los controles heredados:

```
RUN CONTROLS
  COMPRESSION CRITERION = 0.001
  COMPRESSION COUNT = 5
```

`COMPRESSION COUNT` fija el intervalo, en pasos de integración, tras el cual se intenta comprimir un tramo reciente de las historias. `COMPRESSION CRITERION` interviene en el criterio de aproximación que decide si se puede eliminar un punto intermedio.

`CMPAC` considera grupos de tres puntos. Compara el área absoluta de los dos tramos lineales con la aproximación mediante un único tramo entre los extremos. Conserva los puntos cuando la diferencia no cumple el criterio o cuando suprimirlos ocultaría determinados cambios de signo. También trata especialmente amplitudes pequeñas respecto del máximo del historial. Por ello el criterio no debe describirse como una cota directa del error final de tensión o corriente del circuito.

Para evaluar el efecto de la compresión se deben comparar corridas con y sin ella usando la misma red, los mismos núcleos y el mismo control de paso. Cambiar al mismo tiempo el paso impide atribuir las diferencias sólo a esta opción.
Fuentes: `sceptre2novender/nuevo/sceptreDU/cmpac.f90` y `sceptre2novender/nuevo/sceptreDU/cvyclc.f90`.

E.5. Derivadas parciales de parámetros definidos

Origen: sintaxis heredada no desarrollada en el manual traducido. El analizador `DPPRS` del árbol `cuasioriginal` ya contiene el tratamiento de las derivadas parciales. `jacimplic.d` es un caso de prueba nuevo de ese camino; no introduce una palabra clave nueva.

La declaración temporal ordinaria define una ecuación diferencial:

```
DPA = EQUATION 1(PA)
```

La forma con paréntesis a la izquierda proporciona una contribución de derivada parcial utilizada al construir el jacobiano:

```
DPA(PB) = EQUATION 2(PB)
```

La primera identifica la ecuación de evolución de `PA`. La segunda identifica la entrada asociada a esa ecuación y al parámetro `PB`; no llama a `DPA` como si fuera una función ni declara otra condición inicial.

En `jacimplic.d` aparecen ambas formas, además de `DPB`, valores iniciales de `PA` y `PB`, e `INTEGRATION ROUTINE = IMPLICIT`. El programa comprueba mediante `CKPDPD` que existan las derivadas de parámetros a las que se refiere la declaración parcial.

`JDPDEQ` agrega el valor declarado a la entrada correspondiente de la matriz del jacobiano. Esta suma es importante: no se debe afirmar sin más que la declaración reemplaza toda derivada calculada por otros caminos. Tampoco se debe confundir con `GP...`, el diferencial total usado por los análisis de sensibilidad y tolerancias.

El archivo del corpus ejercita la preparación de esa contribución; su presencia no demuestra por sí sola que una fórmula parcial sea matemáticamente coherente con la ecuación diferencial. En un modelo físico, quien proporciona la contribución debe justificarla respecto de la formulación y del jacobiano usados.
Fuentes: `DPPRS`, `CKPDPD`, `JDPDEQ`; `sceptre2novender/corpus/jacimplic.d`.

E.6. Control explícito de la notación vectorial

Origen: control del SCEPTRE de Novender. `VECTOR EQUATIONS` permite elegir el umbral para emplear la formulación vectorial. Los ejemplos pequeños permiten recorrer ese camino sin construir redes enormes.

```
RUN CONTROLS
  VECTOR EQUATIONS = 1
```

El número actúa como umbral de cantidad de elementos, no como un número de ecuaciones que el usuario escribe. En la preparación del árbol de partida, la formulación vectorial se selecciona cuando $N_{\text{elementos}} > N_{\text{máx}}$. Con umbral 1, `a01vec.d` activa ese camino para el circuito de `a01.d`. `vectorial.d` y `vecamp.d` usan 3; `paralelo1200n.d` usa 5000 para conservar la formulación con nombres en una red menor que ese umbral.

El árbol de partida toma normalmente 100 como umbral y ajusta a 70 el caso correspondiente al cálculo de condiciones iniciales cuando no se cambió ese valor. El dato del manual sobre 70 no describe por sí solo todas las situaciones del programa.

La formulación vectorial cambia los nombres del Fortran generado por referencias a arreglos. Los parámetros de una `EQUATION` deben recibirse expresamente como argumentos; depender de nombres libres puede fallar después del renombrado.
Fuentes: `sceptre2novender/nuevo/sceptreM/setup.f90`, `sceptre2novender/nuevo/sceptreM/common_tables.f90`.

E.7. X PLOT DIMENSION y presentación en el terminal

Origen: instrucción y monitor del SCEPTRE de Novender. El anexo original enumera las dimensiones de dibujo; aquí se explica su uso en el monitor de SC1/SD2.

Ejemplo del control del monitor:

```
RUN CONTROLS
  X PLOT DIMENSION = 1
```

En `INTPLT`, cualquier valor no nulo activa un monitor de las variables de salida durante la fase 2. Con `X PLOT DIMENSION = 0` queda desactivado. El valor 1 no significa que el circuito tenga una sola salida ni reduce sus muestras a un punto.

El monitor escribe nombres y valores usando secuencias de control de terminal VT100, como las que entiende habitualmente un terminal de tipo XTerm. En una redirección a un archivo pueden verse esas secuencias en lugar de una tabla de texto convencional. En AC presenta las componentes de los valores complejos.

Este monitor no equivale a las peticiones `PLOT` de `OUTPUTS`, ni el código de `INTPLT` usa `Y PLOT DIMENSION` para seleccionar sus columnas de pantalla. La tabla histórica de dimensiones y este monitor deben explicarse por separado.
Fuentes: `sceptre2novender/nuevo/sceptreDU/intplt.f90`; defecto 56 de `sceptre2novender/pasos/DEFECTOS`.

E.8. FULIST: listados internos de diagnóstico

Origen: instrucción heredada que el manual sólo enumera. No es una nueva opción del lanzador ni una petición de variables de salida.

Ejemplo heredado, `c13.d`:

```
RUN CONTROLS
  FULIST
  RUN INITIAL CONDITIONS ONLY
  RUN SENSITIVITY
```

`FULIST` activa un indicador de diagnóstico de la fase de preparación. En particular, `CKSENS` comprueba ese indicador al preparar las redes adjuntas y las relaciones necesarias para los análisis de sensibilidad y tolerancias.

Entre los listados que puede producir ese camino están:

- `LISTING OF ADJELS`: elementos de la construcción adjunta.
- `LISTING OF IADD`: índices de contribuciones añadidas, si existen.
- `LISTING OF COMMON /ELDATA/`: nombres, nodos, valores e información interna de elementos.
- `LISTING OF SENS` y `LISTING OF ISRS`: tablas internas de sensibilidad.

No todos esos bloques aparecen en cualquier circuito: dependen del análisis solicitado y de las estructuras que éste necesite. El formato es de diagnóstico de la implementación, no un formato estable de intercambio de resultados. Un número de posición interno no debe interpretarse como el nombre de un nodo o como una muestra de la simulación sin consultar la cabecera correspondiente.

`FULIST` no sustituye a `OUTPUTS`, no solicita por sí mismo un análisis de sensibilidad y no equivale a `WRITE DEBUG`. Para un usuario que sólo necesita resultados físicos, normalmente no es necesario activarlo.
Fuentes: `MODULE_RUNCON`, `COMMON_CNTRLS` y `CKSENS`, comparados en los árboles `cuasioriginal` y `nuevo`.

E.9. WRITE DEBUG y los listados de preparación

Origen: instrucción heredada no explicada suficientemente en el manual. La prueba `listados.d` es nueva; reúne controles existentes para comprobar sus caminos de diagnóstico.

```
RUN CONTROLS
  LIST NODE MAP
  WRITE DEBUG
  INTEGRATION ROUTINE = TRAP
  STOP TIME = 0.005
```

`WRITE DEBUG` activa el indicador `DEBUG` de la fase 1. Entre sus consumidores se encuentran la preparación de ecuaciones y la generación del programa. Por ejemplo, `SETUP` puede volcar la lista de funciones mediante `DMPLST`; `GENER8` contiene la presentación de tablas internas de elementos, parámetros, derivadas, valores, tablas y controles.

El contenido sirve para relacionar la entrada con su representación interna. Las cabeceras identifican estructuras como `VALIST`, `TABLST` o `CNTRL TABLE`; no son nuevas secciones que deban copiarse al archivo `.d`. Algunas vistas incluyen posiciones, códigos o representaciones hexadecimales que requieren consultar las estructuras del programa.

`LIST NODE MAP`: solicita el mapa topológico de nodos y ramas.

`WRITE DEBUG`: solicita diagnóstico de estructuras de preparación.

`WRITE SIMUL8 DATA`: permite inspeccionar el Fortran generado; no es sinónimo del control anterior.

Las tres funciones pueden ser útiles al investigar un error, pero ninguna convierte por sí sola un circuito inválido en una simulación válida. Tampoco añaden columnas físicas a la salida numérica.

Los nombres de los archivos de listado los administra el lanzador. En el flujo actual suelen conservarse como `caso.ls1` y `caso.ls2`; no son palabras clave de SCEPTRE.
Fuentes: `SETUP`, `GENER8`, `DMPLST` y `MODULE_RUNCON` del árbol `nuevo`.

E.10. DERIVATIVE como forma larga de DE y DJ

Origen: sintaxis heredada. El ejemplo `a02.d` ya usa la forma larga; la documentación de esta equivalencia completa la exposición del lenguaje, no describe una ampliación local.

```
ELEMENTS
  E1, 1-2 = TABLE 1(TIME)
  DERIVATIVE E1 = TABLE DE1
```

La segunda instrucción declara la derivada temporal de la fuente de tensión `E1`. Su forma abreviada es:

```
  DE1 = TABLE DE1
```

Análogamente, la derivada de una fuente de corriente `J1` se identifica mediante `DJ1`. En una expresión como la anterior, la primera `DE1`, a la izquierda del igual, identifica una derivada de fuente; la situada después de `TABLE` es el nombre de la tabla elegida por el usuario. La coincidencia de grafías no convierte ambas cosas en la misma declaración.

En el ejemplo heredado, `TABLE 1` define la fuente y `TABLE DE1` define su pendiente por tramos. Los tiempos repetidos permiten describir cambios de pendiente. Proporcionar una tabla no hace que el programa compruebe que su contenido sea la derivada matemática de la otra: deben ser coherentes.

La necesidad de derivadas de fuentes depende de la topología. El manual ya explica las mallas de fuentes de tensión con capacidades y los cortes de fuentes de corriente con inductancias; aquí se precisa cómo escribir la declaración larga empleada por el corpus. No debe confundirse con una `DP...` de parámetro definido ni con una diferencial de sensibilidad.
Fuentes: `sceptre2novender/corpus/a02.d`; analizador `ELMPRS`, presente en `cuasioriginal` y `nuevo`.

E.11. Formatos binarios heredados de resultados

La fase 2 escribe resultados binarios por la unidad asociada a `fort.7`; el lanzador administra nombres como `caso.dirac`. Hay que distinguir ese binario de la salida de texto `.sim` y reconocer dos disposiciones heredadas.

POS antiguo: Arreglos separados de 51 nombres, tipos, longitudes, direcciones y reservas; firma inicial `DIRECTOR` y fin `ZZZZZZZZ`. Es un antecedente de los árboles 00.317, 2012.061 y 2013-0.6.

POS4 de SC1/SD2: Entradas consecutivas de 32 bytes en un único registro de 2048 bytes. La cabecera `DIRECTORY` indica `recno=1`.

En el formato POS4 de SC1/SD2, cada entrada ocupa:

Desplazamiento, desde cero	Longitud	Campo
0–11	12 bytes	Nombre del vector.
12–22	11 bytes	Reserva; fecha en la cabecera.
23	1 byte	Tipo.
24–27	4 bytes	Longitud en elementos.
28–31	4 bytes	Registro de comienzo de los datos.

En la primera entrada, de tipo 0, la longitud vale 2048 y el último campo vale 1: el directorio ocupa un solo registro. En las entradas de datos, `recno` sigue siendo una dirección absoluta: el desplazamiento de lectura es $(\texttt{recno} - 1)\,2048$. El tipo 4 identifica números `REAL*4`; el tipo 9 permite texto, como `HEADING`; el tipo 1 indica un bloque libre y 255, interpretado como -1, marca el final libre del directorio.

Un directorio de un registro tiene 64 posiciones contando la cabecera. No equivale a 63 salidas físicas disponibles: también ocupan entradas el eje, los títulos y las marcas correspondientes. Esta explicación describe el formato recibido de Novender, con directorio de un registro.

Fuentes: árbol `cuasioriginal`, rutinas `POS4_DIR`, `POS4_INTERFACE` y `POST`; formatos anteriores conservados en el repositorio.

E.12. Funciones Fortran del corpus: dependencias y salidas propias

Origen: mecanismo y funciones auxiliares del SCEPTRE de Novender. Una llamada en el mazo no contiene necesariamente la implementación necesaria para enlazar. También hay que conocer los efectos laterales de las funciones de usuario.

Caso	Funciones	Qué hay que proporcionar
`g02.d`	`FSAT`	Función de saturación; existe fuente `g02.f90` junto a los ejemplos distribuidos.
`c10.d`	`FTABL1`, `FALL`, `FIZK`, `FTOR`	Rutinas auxiliares del inversor, que deben estar disponibles al enlazar.
`i01.d`	`FMOS`	Modelo MOS externo, que debe estar disponible al enlazar.
`c11.d`	`FC11`	Rutina de registro de la cadena de solitones; recibe tiempo y 97 tensiones en esta variante.

`c12.d`	`FC12`	Rutina que reúne filas de una malla y escribe resultados auxiliares.

En `c11.d`, el parámetro `PDUMM` llama a `FC11`. Su fuente abre la unidad Fortran 2, escribe el tiempo y las tensiones y devuelve `TIME`. Por tanto, su finalidad no se deduce del valor escalar de `PDUMM`: produce además un flujo de texto independiente. `OPEN(2,...)` no fija un nombre mediante `FILE=`; no debe presentarse ese flujo como una serie Dirac ni como la salida textual normal de SCEPTRE. Su destino depende del entorno de ejecución Fortran.

- La cantidad, el orden y el tipo de los argumentos deben coincidir entre la llamada del mazo y la función enlazada. En `c11.d`, la lista continúa en varias líneas; no es una función de un argumento vectorial.
- Una función llamada durante la evaluación del modelo puede ejecutarse en etapas internas y en evaluaciones repetidas de un mismo instante. Sus escrituras no deben interpretarse automáticamente como un registro por paso aceptado o por intervalo de salida.
- `FC12` conserva información entre llamadas. Su rama para índice negativo retorna sin asignar explícitamente el resultado: ese camino no debe usarse como ejemplo de una función numérica correctamente definida en todos los casos.

El manual del núcleo y el contrato de flujos no bastan para describir los archivos que puede abrir una función de usuario. Al documentar o aislar una ejecución hay que revisar también esas funciones y sus unidades de entrada/salida.
Fuentes: `corpus/corpus/novender/c11.d`, `corpus/corpus/novender/c11.f90`, `corpus/corpus/novender/c12.f90`; `sceptre2novender/usrlib/README.md`.

E.13. Condiciones iniciales incompatibles en lazos y cortes

Origen: limitación observada del tratamiento heredado de la topología. El manual explica las restricciones de lazos capacitivos y cortes inductivos. Falta relacionarlas con los mazos del corpus que solicitan estados iniciales mutuamente incompatibles y con su comparación con Siru.

Caso capacitivo: `dos-cap-salto.d`, con variantes `cap-C-salto.d`, `cap-CB-salto.d` y `cap-Q.d`. Dos capacitores de valores 1 y 2 están en paralelo; se declaran `VC1=0` y `VC2=1`. La conexión exige

$$v_{C1}(0) = v_{C2}(0).$$

Por tanto, esas dos condiciones no describen un estado inicial admisible de la red ya conectada. El informe del corpus registra que SCEPTRE seleccionó un estado independiente y arrancó con ambas tensiones nulas en el caso examinado; no realizó el equilibrio por conservación de carga empleado por Siru. Este resultado registrado no es una regla que permita predecir qué condición conservará cualquier ordenamiento de una red.

Si se quiere modelar la conexión de dos capacitores previamente aislados, la conservación de carga da, bajo esas hipótesis físicas,

$$v(0^+) = \frac{C_1 v_1(0^-) + C_2 v_2(0^-)}{C_1 + C_2} = \frac{2}{3}.$$

Eso supone un acontecimiento de conexión anterior al estado integrado. No se obtiene simplemente escribiendo dos tensiones incompatibles en `INITIAL CONDITIONS`.

Caso inductivo: `dos-bobinas-salto.d` conecta dos bobinas en serie con `IL1=0` e `IL2=1`. En el nodo común, sin otra rama, la ley de corrientes exige una única corriente de serie. También aquí hay que distinguir la red ya conectada de un proceso impulsivo de conexión; SCEPTRE no debe suponerse un resolvedor automático del trasvase de flujo descrito en los comentarios sobre Simul.

1. Verificar que los valores iniciales satisfagan las restricciones algebraicas de la topología.
2. Para comparar integradores, proporcionar el mismo estado físico compatible en ambos programas.
3. Si se estudia una conmutación, describir por separado la regla de equilibrio o el modelo que resuelve el acontecimiento.
4. No atribuir a SCEPTRE un equilibrio que el caso examinado no efectúa: aquí se documenta el comportamiento registrado, sin modificar los mazos.

Los nombres `cap-C`, `cap-CB` y `cap-Q` identifican la procedencia de las variantes. En los archivos citados, las ramas escritas para SCEPTRE siguen siendo `C1` y `C2`; el nombre del archivo o sus comentarios no seleccionan un tipo de capacitor de Siru.
Fuentes: `corpus/README.md`, apartado de condiciones iniciales desparejas; `corpus/corpus/corrida/dos-cap-salto.d`; `corpus/corpus/corrida/dos-bobinas-salto.d`; `corpus/SCEPTRE2.md`. Son observaciones registradas; no se repitieron las simulaciones durante esta revisión documental.

E.14. Interpretar los valores no lineales de C y L en los ejemplos

Origen: aplicación de funciones heredadas; complemento práctico de la teoría del manual. Los ejemplos `B07`, `B16` y `V050` requieren distinguir una capacidad o inductancia diferencial de un cociente total. La dependencia no lineal no identifica por sí sola cuál de ambas magnitudes se ha tabulado.

En `beispiele/b07.d`, la inductancia se obtiene de una tabla evaluada con `PNI=10.*IL1`:

```
ELEMENTS
  L1, 2-3 = Q1(100., TABLE 1(PNI))
DEFINED PARAMETERS
  PNI = X2(10.*IL1)
FUNCTIONS
  Q1(A,B) = (A*B)
```

Es una composición de tabla y ecuación del lenguaje original, no una instrucción especial para saturación. La fuente del material, la escala de corriente y las unidades elegidas forman parte del modelo físico.

Para una bobina con flujo concatenado $\lambda(i)$, la inductancia diferencial es $L_d(i) = \mathrm{d}\lambda/\mathrm{d}i$ y la relación es $v = L_d(i)\,\mathrm{d}i/\mathrm{d}t$. Si se parte de una inductancia total $L_t(i) = \lambda(i)/i$, la conversión correspondiente es

$$L_d(i) = L_t(i) + i\frac{\mathrm{d}L_t}{\mathrm{d}i}.$$

En `beispiele/v050.d`, `PL=PMU*PS`, `PMU=TABLE MUEH(PHfe)` y `PHfe=IL*PN` construyen el valor de `L` a partir del campo. El parámetro `PHI`, con `DPHI=VL`, integra además la tensión. La presencia de ese parámetro no hace que el programa convierta automáticamente una

tabla de permeabilidad total en diferencial: debe revisarse qué magnitud proporciona la tabla.

`beispiele/b16.d` compara ramas con capacidades constantes y variables en el tiempo con parámetros definidos integrados. Para un capacitor, $i = C_d(v)\,\mathrm{d}v/\mathrm{d}t$ y $q = C_t(v)v$ sólo son equivalentes tras convertir correctamente C_t en C_d. Si la dependencia incluye explícitamente tiempo u otra variable, la derivada total de la carga puede introducir términos adicionales; no basta la conversión univariable.

- Los nombres `CT`, `PC` o `PL` no son declaraciones de capacidad total ni de inductancia diferencial.
- Las tablas con abscisas repetidas pueden producir cambios bruscos de pendiente y exigir revisar el paso y la convergencia.
- Que dos programas lean valores numéricos iguales no garantiza que hayan recibido la misma ley constitutiva.

Fuentes: `corpus/corpus/beispiele/b07.d`, `corpus/corpus/beispiele/b16.d`, `corpus/corpus/beispiele/v050.d`; capítulos 2 y 5 del manual.

E.15. Ejemplos dependientes de modelos permanentes y modelos de SPICE

Origen: modelos y mecanismo de biblioteca heredados; aclaración de las dependencias del corpus. El manual describe `MODEL`, `TEMP` y `PERM`. Los casos generales muestran además dos problemas de reproducción: un modelo permanente puede estar ausente de la biblioteca utilizada o ya estar instalado cuando el mazo intenta definirlo otra vez.

- `beispiele/b03.d` declara un modelo `1N754` permanente y lo utiliza en el circuito. Para reproducir el ejemplo se necesita conocer el estado inicial de la biblioteca, no sólo conservar el archivo `.d`.
- `beispiele/c18.d` declara `SPICE_SCR(PERM)`; `c19.d` invoca ese modelo permanente. La disponibilidad del segundo depende de la biblioteca con la que se ejecute.
- Otros ejemplos de rectificadores llaman a modelos no definidos en el propio mazo. Una referencia sin resolver no se repara eligiendo otro integrador.

En `c18.d`, el nombre `SPICE_SCR` identifica un equivalente de tiristor expresado mediante elementos y tablas de SCEPTRE:

```
RG, G-K = 20
RS, A-K = TABLE RS(VCR)
CR, S-E = 10E-6
RE, S-E = 1
JE, E-S = X2(50*IRG + 11*IRS)
JC, S-E = TABLE CL(VCR)
```

El circuito usa `MODEL SPICE_SCR(PERM,CHANGE JC=0)`. El cambio afecta esa instancia. Que el modelo se base en una publicación de PSPICE no significa que el lector de SCEPTRE acepte una netlist de SPICE ni sus instrucciones `.MODEL`.

1. Identificar si el mazo define el modelo o espera encontrarlo instalado.

2. Registrar la biblioteca empleada y conservar las definiciones auxiliares publicadas.
3. Distinguir un error de biblioteca o de enlace de un error de integración.
4. Al preparar una variante autocontenida, conservar la procedencia y diferenciarla del listado extraído del libro.

El registro de extracción del 6 de octubre encontró definiciones permanentes redundantes y modelos ausentes. Es un diagnóstico de aquella biblioteca y ejecución; no debe repetirse como estado actual universal.
Fuentes: `corpus/corpus/beispiele/b03.d`, `corpus/corpus/beispiele/c18.d`, `corpus/corpus/beispiele/c19.d`; `corpus/corpus/beispiele/README.md`; capítulo 2, descripción de modelos.

E.16. D01: filtro compuesto, salida compleja y repeticiones AC

Origen: ejemplo heredado; explicación complementaria. `d01.d` estudia en frecuencia un filtro compuesto con elementos no ideales. No es una integración transitoria y no contiene la orden `COMPUTER TIME LIMIT`. El prefijo `D01` del título identifica el ejemplo; no declara una derivada ni selecciona un tipo de análisis.

```
CIRCUIT DESCRIPTION
D01: Composite filter (AC analysis) using nonideal elements
ELEMENTS
  EAC,  A - GND = (1,0)
  RI,   A - 1 = 0.1
  R2,   1 - GND = 1E6
  L3,   1 - 2 = 1.414E-3
  R4,   2 - 3 = 25
  C5,   3 - GND = 0.0795E-6
  L6,   1 - 4 = 2.12E-3
  R7,   4 - 5 = 30
  C8,   5 - GND = 0.265E-6
  L9,   5 - 6 = 2.0075E-3
  R10,  6 - 7 = 25
  L11,  7 - 8 = 0.946E-3
  R12,  8 - 9 = 29
  C13,  9 - GND = 0.1365E-3
  L14,  7 - 10= 1.4775E-3
  R15,  10- 11= 29
  L16,  11- 12= 1.414E-3
  R17,  12- 13= 25
  C18,  13- GND = 0.0795E-6
  RL,   11- GND = 50
DEFINED PARAMETERS
  WVRL = X1 (VRL)
  WDB  = X2 (20*LOG10(ABS(VRL/EAC)))   ; 05/10/2026: tenia un ) de mas (SCEPTRE lo acepta,
     ExprTk no)
OUTPUTS
  VRL, COMPLEX
  WVRL,DEGREES
  WDB,COMPLEX
RUN CONTROLS
  RUN AC
  PRINT EIGENVALUES
  INITIAL FREQUENCY = 1000
  FINAL FREQUENCY = 50E3
```

```
  NUMBER FREQUENCY STEPS = 50
  TYPE FREQUENCY RUN = LOG
  X PLOT DIMENSION = 0
RERUN DESCRIPTION (2)
ELEMENTS
  RL = 100, 200
END
```

Este listado contiene el mazo completo: puede entregarse a SCEPTRE. La procedencia y la corrección de la expresión se explican al final de la sección.

1. `EAC=(1,0)` es una excitación fasorial unitaria. La declaración de `RUN AC` hace que se resuelva el sistema complejo para cada frecuencia.

2. El barrido es logarítmico, desde 1000 hasta 50000 en las unidades de frecuencia coherentes del ejemplo. El número indicado controla los pasos del barrido; no son pasos de integración temporal.

3. `VRL,COMPLEX` solicita las componentes de la tensión de carga; `WVRL,DEGREES` presenta la misma magnitud compleja en la modalidad de amplitud y fase angular del análisis AC.

4. `WDB` calcula $20 \log_{10} |V_{RL}/E_{AC}|$. Se usa 20 por tratarse de una relación de amplitudes de tensión. El prefijo `W` permite definir parámetros del cálculo complejo; no designa potencia por su nombre.

5. La maestra usa `RL=50`; las dos repeticiones la sustituyen sucesivamente por 100 y 200, conservando las instrucciones de análisis aplicables de la maestra.

6. `PRINT EIGENVALUES` solicita información adicional del sistema; `X PLOT DIMENSION=0` desactiva el monitor inmediato. Ninguna de las dos órdenes cambia el filtro.

Variantes del corpus. La copia en `corpus/corpus/novender/d01.d` conserva un exceso de paréntesis al final de la expresión de `WDB`. En el listado de esta explicación se escribe la expresión equilibrada; la corrección también está registrada en la copia de extttd01.d del corpus del proyecto. Esta diferencia debe conservarse al identificar la procedencia: no conviene reproducir el paréntesis sobrante como parte de la sintaxis válida. Los registros antiguos de dificultades con funciones complejas son diagnósticos de la versión ejecutada, no una propiedad física del ejemplo.
Fuentes: `corpus/corpus/novender/d01.d`, `corpus/corpus/beispiele/d01.d`, `sceptre2novender/corpus/d01.d`; capítulo 2 del manual, análisis AC, parámetros definidos, salidas y repeticiones.

F Maestría en Energías Renovables de la Universidad Nacional de Salta

Este apéndice presenta el contexto académico solicitado para esta edición. La Universidad Nacional de Salta, mediante su Facultad de Ciencias Exactas, ofrece la Maestría en Energías Renovables, orientada a la formación y la investigación sobre el aprovechamiento energético y la transición hacia fuentes renovables. *Transformaciones Energéticas* es una asignatura de su trayecto común; no es el nombre de otra maestría.

F.1. Marco institucional y resoluciones

Documento	**Identificación y alcance**
Plan de estudios	R-CS-2021-0135, del 27 de mayo de 2021.
Reglamento	R-CDEXA-2021-048, ratificada por R-CS-2021-0133.
Acreditación	RESFC-2022-164-APN-CONEAU#ME, categoría B, según la ficha oficial de la carrera.
Reconocimiento ministerial	Resolución 601/23, informada en el catálogo de posgrados de la UNSa.
Inicio de ciclo	Resolución CDEXA 148/24: inicio anunciado para julio/agosto de 2024.
Curso de Transformaciones Energéticas	RCD-477-2024-EXA-UNSa, del 4 de julio de 2024: autoriza la edición y aprueba su anexo académico.

Fuentes institucionales: ficha de la Facultad de Ciencias Exactas y catálogo de posgrados de la Universidad. El domicilio académico es la sede de Salta de la Facultad. La ficha presenta la carrera como presencial, de dos años; exige titulación de al menos cuatro años y admisión académica.

F.2. Estructura del Plan 2021

El plan fija **720 horas reloj**: 360 de formación común, 200 de orientación y 160 correspondientes a 16 créditos. También requiere tesis. Las orientaciones son bioenergía, energía eólica, arquitectura bioclimática, solar fotovoltaica y solar térmica. La modalidad es presencial, con posibilidad de componentes a distancia hasta el 30 % de las horas.

Trayecto común	**Horas**
Transformaciones Energéticas	40
Ambiente y problemas ambientales	40
Fundamentos de energía solar	40
Fundamentos de bioenergía	40
Fundamentos de energía eólica	40
Políticas energéticas y gestión de proyectos de inversión	30
Sistemas de Información Geográfica aplicados a energías renovables	40
Eficiencia energética y energías renovables	40

Seminario de tesis y práctica profesional	40
Taller de actualización	10
Total	**360**

La organización y las horas se cotejaron en las páginas 10 a 12 del PDF del plan; los contenidos mínimos de Transformaciones Energéticas están en su página 14. El plan prevé cursos intensivos generalmente semanales. Los bloques del calendario de la carrera no deben confundirse con la duración de cada asignatura.

F.3. Transformaciones Energéticas: contenidos mínimos

El mínimo del Plan 2021 abarca **fundamentos termodinámicos para energías renovables y transferencia de calor y masa**. Su desarrollo se concreta en el programa analítico de la resolución del curso.

F.4. Programa analítico de Transformaciones Energéticas

Síntesis del anexo de la RCD-477/2024, página 3 del PDF:

Termodinámica: Trabajo y energía; calor, energía interna y entalpía; balances abiertos; leyes termodinámicas, entropía, irreversibilidad y exergía; gases ideales, Carnot, agua-vapor, Rankine y generación térmica.

Conducción: Conductividad y ecuación del calor; resistencias y analogía eléctrica; paredes y geometrías radiales, aletas; regímenes estacionarios y transitorios, Biot, Fourier, capacidad concentrada y difusión de masa.

Convección: Capas límite, semejanza dimensional, Reynolds, Nusselt y Prandtl; regímenes laminar/turbulento, flujos internos/externos y determinación experimental del coeficiente térmico.

Radiación: Espectro e intensidad, ángulo sólido, irradiación y radiosidad; cuerpo negro, Planck, Wien y Stefan–Boltzmann; emisividad, cuerpo gris, absorción, reflexión, transmisión y factores de forma.

Laboratorios: Conducción en barras, convección en paredes y radiación.

La resolución establece 40 horas presenciales, del 29 de julio al 2 de agosto de 2024, bajo dirección de Ester Sonia Esteban, con clases, problemas, laboratorio y examen escrito. La bibliografía completa figura en las páginas 3 y 4 de la resolución enlazada.

F.5. Calendario y relación con la simulación

La página institucional consultada reúne un calendario de la cohorte iniciada en 2024, con actividades hasta marzo de 2026. Eso no acredita un dictado de Transformaciones Energéticas en marzo de 2026 ni una nueva inscripción abierta en la fecha de consulta. Para futuras ediciones corresponde consultar la convocatoria de la carrera o escribir a posgrado@exa.unsa.edu.ar.

En solar térmica, el plan incluye simulación de procesos y menciona explícitamente **Simusol**, con elementos, funciones, ecuaciones, almacenamiento térmico, cambio de fase, control y validación experimental (página 19 del PDF). Esta referencia curricular es distinta del programa de Transformaciones Energéticas.

Bibliografía

[1] J. C. Bowers y S. R. Sedore. *SCEPTRE: A Computer Program for Circuit and System Analysis*. Englewood Cliffs, Nueva Jersey: Prentice-Hall, 1971.

[2] D. Becker. *Extended SCEPTRE. User's Manual*. AFWL TR-73-75; NTIS ADA 009594. https://www.thm.de/iem/images/user/novender-978/SCEPTRE-Vol1-en.pdf.

[3] D. Becker. *Extended SCEPTRE. Mathematical Formulation*. AFWL TR-73-75; NTIS ADA 009595. https://www.thm.de/iem/images/user/novender-978/SCEPTRE-Vol2-en.pdf.

[4] Wolf-Rainer Novender. *SC1/SD2 — Revised SCEPTRE Version*. Inédito, 2026.

[5] R. W. Jensen y L. P. McNamee. *Handbook of Circuit Analysis Languages and Techniques*. Englewood Cliffs, Nueva Jersey: Prentice-Hall, 1976.

[6] *Advanced Statistical Analysis Program (ASTAP). General Information Manual*. IBM GH20-1271-0.

[7] H. Spiro. *Simulation integrierter Schaltungen durch universelle Rechnerprogramme. Verfahren und Praxis der rechnergestützten Simulation nichtlinearer Schaltungen*. Múnich: Verlag Oldenbourg, 1985.

[8] W.-R. Novender. *SCEPTRE — Ein Programm zur Analyse und Simulation nichtlinearer elektrischer Netzwerke und dynamischer Systeme. Benutzerhandbuch*. 2018. https://www.thm.de/iem/images/user/novender-978/SCEPTRE-Vol12-de.pdf.

[9] W. Kaplan. *Ordinary Differential Equations*. Reading, Massachusetts: Addison Wesley, 1958, cap. 10-2, pp. 400–401.

[10] W. Kaplan. *Ordinary Differential Equations*. Reading, Massachusetts: Addison Wesley, 1958, cap. 10-4, «Heun's Method», pp. 402–403.

[11] K. G. Ashar, H. N. Ghosh, A. W. Aldridge y L. J. Patterson. «Transient Analysis and Device Characterization of ACP Circuits». *IBM Journal of Research and Development* 7 (1963), p. 218.

[12] P. E. Chase. «Stability Properties of Predictor-Corrector Methods for Ordinary Differential Equations». *Journal of the ACM* 9 (octubre de 1962), pp. 457–468.

[13] Wm. C. Davidon. *Variable Metric Method for Minimization*. Informe ANL-5990 (REV). AEC Research and Development, 1959.

[14] M. J. Box. «A Comparison of Several Current Optimization Methods, and the Use of Transformation in Constrained Problems». *The Computer Journal* 9 (1966), pp. 67–77.

[15] M. J. Box, D. Davies y W. G. Svann. *Non-Linear Optimization Techniques*. Monograph No. 5. Imperial Chemical Industries Ltd., 1969.

[16] *Variable Metric Minimization*. SHARE Routine No. 1117, AN Z013, A3. SHARE Inc., 25 Broadway, Suite 750, Nueva York, NY 10004.

[17] W. Tinney y J. Walker. *Direct Solution of Sparse Network Equations by Optimally Ordered Triangular Factorization*. IEEE, noviembre de 1967.

[18] J. Grad y M. A. Brebner. «Algorithm 343, Eigenvalues and Eigenvectors of a General Matrix». *Communications of the ACM* 11 (diciembre de 1968), pp. 820–826.

[19] J. V. Leeds y G. I. Ugron. «Simplified Multiple Parameter Sensitivity Calculation and Continuously Equivalent Network». *IEEE Transactions on Circuit Theory* 14 (junio de 1967).

[20] W.-R. Novender. *SCEPTRE — Ein Programm zur Analyse und Simulation nichtlinearer elektrischer Netzwerke und dynamischer Systeme. Anwendungsbeispiele (deutsch/englisch)*. 2018. https://www.thm.de/iem/images/user/novender-978/SCEPTRE-beispiele.pdf.

[21] J. Portasik y C. Glass. *System Analysis Using the SCEPTRE Computer Program*. Informe 98. AFWL, septiembre de 1968.

[22] S. R. Sedore. «SCEPTRE: A Program for Automatic Network Analysis». *IBM Journal of Research and Development* 2 (noviembre de 1967).

[23] S. R. Sedore. «More Efficient Use of the F Matrix in Practical Circuit Analysis Program». *IEEE Transactions on Computers* 17 (mayo de 1968).

[24] M. Fowler y R. Warten. «A Numerical Integration Technique for Ordinary Differential Equations with Widely Separated Eigenvalues». *IBM Journal of Research and Development* 2 (septiembre de 1967).

[25] R. H. Dickhaut. *Advances in Computer Techniques for Radiation Effects Calculation*. D2-90571. Boeing, 1964.

[26] W. C. Cordwell. *Radiation Device Modeling for the SCEPTRE Program*. 69-825-2371. Owego, Nueva York: IBM, junio de 1969.

[27] M. B. Reed y S. Seshu. *Linear Graphs and Electrical Networks*. Reading, Massachusetts: Addison-Wesley, 1961.

[28] F. B. Hildebrand. *Introduction to Numerical Analysis*. Nueva York: McGraw-Hill, 1956.

[29] R. M. Warten. «Automatic Step-Size Control for Runge-Kutta Integration». *IBM Journal of Research and Development* 7(4) (octubre de 1963), pp. 340–341.

[30] C. W. Gear. «The Numerical Integration of Ordinary Differential Equations». *Mathematics of Computation* 21 (1967).

[31] C. W. Gear. «The Automatic Integration of Ordinary Differential Equations». *Communications of the ACM* 14 (marzo de 1971).

[32] S. W. Director y R. A. Rohrer. «The Generalized Adjoint Network and Network Sensitivities». *IEEE Transactions on Circuit Theory* 16 (agosto de 1969).

[33] S. W. Director y R. A. Rohrer. «Automated Network Design – The Frequency-Domain Case». *IEEE Transactions on Circuit Theory* 16 (agosto de 1969).

Siglas institucionales del original: AFWL: Air Force Weapons Laboratory, Kirtland AFB, Nuevo México 87117, EE. UU.; NTIS: National Technical Information Service, Springfield, Virginia 22151, EE. UU.; AFAPL: Air Force Aero Propulsion Laboratory, Wright-Patterson AFB, Ohio 45433, EE. UU.

Índice alfabético

Actuador electrohidráulico, 66
Análisis AC, 18, 91
Análisis DC, 16
Análisis transitorio, 18, 72

Darlington, circuito, 48
Davidon, método de, 90

Ebers–Moll, modelo de, 105
Ecuaciones diferenciales, 52
Ejemplos
    interdisciplinarios, 66
Elementos
    de circuito, 9

Formato de entrada, 8
Fuentes
    derivadas temporales, 11
Funciones, 15
    FPULSE, 38
    FSINSQ, 39
    periódicas, 37

Inversor, 42

LU, factorización, 111

Matrices dispersas, 111
Modelos, 21
Monte Carlo, 20, 54, 88
Métodos de integración, 31, 77
    exponencial, 80
    implícitos, 36, 82
    Runge–Kutta, 77
    trapecio modificado, 78

Newton–Raphson, 84

Optimización, 20, 61, 90

Parámetros definidos, 12
Peor caso, 20, 59, 89
Punto de trabajo, 35

Retardo computacional, 31

Seguidor de emisor, 45
Sensibilidad, 20, 56, 88

Tablas, 15
Tolerancias
    límites, 11
Transistor
    circuitos equivalentes, 105
    pequeña señal, 50

Variables de estado, 34, 92