

# BCL

Librerías Estándar  
Referencia Completa

Versión 2.0.0

*BCL Development Team*

Noviembre 2025

**BCL Standard Libraries Reference**  
Version 2.0.0

Copyright © 2025 BCL Development Team

Este documento describe las librerías estándar del lenguaje de programación BCL.

Para más información, visite:  
<https://github.com/yourusername/bcl>

# Índice general

<b>1. Introducción</b>	<b>5</b>
1.1. Acerca de este documento . . . . .	5
1.2. Requisitos . . . . .	5
1.2.1. Versión de BCL . . . . .	5
1.2.2. Terminal compatible . . . . .	5
1.3. Instalación . . . . .	6
1.4. Uso básico . . . . .	6
1.4.1. Cargar una librería . . . . .	6
1.4.2. Arrays globales en procedimientos . . . . .	6
1.5. Convenciones . . . . .	7
1.5.1. Nomenclatura . . . . .	7
1.5.2. Parámetros . . . . .	7
1.5.3. Códigos de ejemplo . . . . .	7
1.6. Estructura del documento . . . . .	7
1.7. Soporte y contribuciones . . . . .	8
1.8. Licencia . . . . .	8
<b>2. ANSI - Control de Terminal</b>	<b>9</b>
2.1. Introducción . . . . .	9
2.2. Control de Pantalla . . . . .	9
2.3. Colores . . . . .	9
2.3.1. Constantes de Color . . . . .	9
2.3.2. Uso . . . . .	9
2.4. Caracteres Unicode v2.0 . . . . .	10
2.4.1. Box Drawing - Línea Simple . . . . .	10
2.4.2. Box Drawing - Línea Doble . . . . .	10
2.4.3. Box Drawing - Redondeado . . . . .	10
2.4.4. Otros Símbolos . . . . .	10
2.5. Ejemplo: Dibujar Caja . . . . .	10
<b>3. WINDOW - Gestión de Ventanas</b>	<b>13</b>
3.1. Introducción . . . . .	13
3.2. Creación y Dibujo . . . . .	13
3.3. Contenido . . . . .	13
3.4. Elementos Interactivos . . . . .	14
3.4.1. Menú . . . . .	14
3.4.2. Barra de Progreso . . . . .	14
3.4.3. Botones . . . . .	14
3.5. Contenido Scrollable . . . . .	14

3.6. Gestión de Ventanas . . . . .	14
3.7. Ejemplo Completo . . . . .	15
<b>4. MATRIX - Operaciones Matriciales</b>	<b>17</b>
4.1. Introducción . . . . .	17
4.2. Creación de Matrices . . . . .	17
4.3. Operaciones . . . . .	17
4.4. Análisis . . . . .	18
4.5. Utilidades . . . . .	18
4.6. Ejemplo Completo . . . . .	18
4.7. Almacenamiento . . . . .	19
<b>5. CALCULUS - Cálculo Numéricico</b>	<b>21</b>
5.1. Introducción . . . . .	21
5.2. Carga y configuración . . . . .	21
5.3. Derivadas Numéricicas . . . . .	22
5.3.1. CALC_DERIV_FORWARD . . . . .	22
5.3.2. CALC_DERIV_CENTRAL . . . . .	22
5.3.3. CALC_DERIV2 . . . . .	22
5.4. Integración Numérica . . . . .	23
5.4.1. CALC_INTEGRATE_TRAP . . . . .	23
5.4.2. CALC_INTEGRATE_SIMPSON . . . . .	23
5.4.3. Comparación de métodos . . . . .	24
5.5. Ecuaciones Diferenciales . . . . .	24
5.5.1. CALC_EULER . . . . .	24
5.5.2. CALC_RK4 . . . . .	24
5.6. Búsqueda de Raíces . . . . .	25
5.6.1. CALC_NEWTON . . . . .	25
5.6.2. CALC_BISECTION . . . . .	26
5.6.3. CALC_SECANT . . . . .	26
5.6.4. Comparación de métodos . . . . .	26
5.7. Polinomios . . . . .	26
5.7.1. CALC_POLY_EVAL . . . . .	26
5.7.2. CALC_POLY_ROOTS_QUAD . . . . .	27
5.8. Series y Límites . . . . .	27
5.8.1. CALC_SERIES_SUM . . . . .	27
5.8.2. CALC_LIMIT . . . . .	27
5.9. Funciones Utilidad . . . . .	28
5.9.1. CALC_FACTORIAL . . . . .	28
5.9.2. CALC_COMBINATION . . . . .	28
5.9.3. CALC_GCD . . . . .	28
5.10. Ejemplo Completo: Análisis de Función . . . . .	28
5.11. Notas de Precisión . . . . .	29
<b>A. Referencia Rápida</b>	<b>31</b>
A.1. Resumen de Librerías . . . . .	31
A.2. Tabla de Funciones por Librería . . . . .	31
A.2.1. ANSI (21 constantes + funciones) . . . . .	31
A.2.2. WINDOW (26 funciones) . . . . .	31
A.2.3. MATRIX (23 funciones) . . . . .	31

A.2.4. CALCULUS (27 funciones) . . . . .	31
A.3. Caracteres Unicode Box Drawing . . . . .	32
A.4. Valores Predeterminados . . . . .	32
<b>B. Ejemplos Adicionales</b> . . . . .	<b>33</b>
B.1. Gráfico de Función con Ventanas . . . . .	33
B.2. Calculadora Matricial Interactiva . . . . .	34
<b>C. Solución de Problemas</b> . . . . .	<b>37</b>
C.1. Errores Comunes . . . . .	37
C.1.1. Arrays no persisten en procedimientos . . . . .	37
C.1.2. Terminal no muestra Unicode . . . . .	37
C.1.3. Métodos numéricos no convergen . . . . .	37
C.2. Optimización . . . . .	38
C.2.1. Performance de Matrices . . . . .	38
C.2.2. Integración Numérica . . . . .	38
C.3. Recursos Adicionales . . . . .	38



# Capítulo 1

## Introducción

### 1.1. Acerca de este documento

Este manual describe las librerías estándar incluidas en BCL (Basic Command Language) versión 2.0 y posteriores. Las librerías estándar proporcionan funcionalidad avanzada para:

- Control de terminal y gráficos ANSI
- Gestión de ventanas en terminal
- Operaciones matriciales estilo MATLAB
- Cálculo numérico y análisis matemático

### 1.2. Requisitos

#### 1.2.1. Versión de BCL

Todas las librerías descritas en este documento requieren BCL versión 2.0.0 o superior. Esta versión incluye mejoras fundamentales:

- **Sistema de arrays globales:** Los arrays ahora funcionan correctamente con la declaración GLOBAL en procedimientos
- **Soporte Unicode completo:** Secuencias de escape \uXXXX y \UXXXXXXXXX para UTF-8
- **Librerías mejoradas:** ANSI y WINDOW completamente reescritas, MATRIX y CALCULUS nuevas

#### 1.2.2. Terminal compatible

Para aprovechar las funcionalidades de ANSI y WINDOW, se requiere un emulador de terminal que soporte:

- Secuencias de escape ANSI
- Codificación UTF-8
- Caracteres Unicode box-drawing
- Colores de 16 bits (8 básicos + 8 brillantes)

Terminales recomendadas: xterm, gnome-terminal, konsole, iTerm2, Windows Terminal.

### 1.3. Instalación

Las librerías estándar se incluyen en la distribución de BCL en el directorio lib/:

```
BCL/
  +- lib/
    |-- ANSI.BLB      # Terminal control y gráficos
    |-- WINDOW.BLB   # Gestión de ventanas
    |-- MATRIX.BLB   # Operaciones matriciales
    +- CALCULUS.BLB  # Cálculo numérico
```

No se requiere instalación adicional. Las librerías están listas para usar mediante el comando **SOURCE**.

### 1.4. Uso básico

#### 1.4.1. Cargar una librería

Para utilizar una librería en un script BCL, use el comando **SOURCE**:

```
1 #!/usr/bin/env bcl
2
3 SOURCE "lib/ANSI.BLB"
4 SOURCE "lib/WINDOW.BLB"
5 SOURCE "lib/MATRIX.BLB"
6 SOURCE "lib/CALCULUS.BLB"
```

#### Nota

Las rutas de las librerías son relativas al directorio de trabajo actual. Si ejecuta scripts desde otros directorios, ajuste la ruta según sea necesario.

#### 1.4.2. Arrays globales en procedimientos

##### Advertencia

**Importante:** Debido a limitaciones actuales de BCL, los arrays utilizados por las librerías MATRIX y algunas funciones de otras librerías deben declararse como **GLOBAL** cuando se usen dentro de procedimientos.

Ejemplo correcto:

```
1 PROC MI_FUNCION DO
2     GLOBAL M A B resultado      # Declarar arrays como globales
3     MAT_ZEROS M 3 3
4     MAT_ADD A B resultado
5 END
```

Ejemplo incorrecto (no funcionará):

```
1 PROC MI_FUNCION DO
2     # Falta GLOBAL - los arrays no persistirán
```

```

3   MAT_ZEROS M 3 3          # ERROR
4   MAT_ADD A B resultado    # ERROR
5 END

```

## 1.5. Convenciones

### 1.5.1. Nomenclatura

Todas las funciones de librería siguen estas convenciones:

- **Prefijo por librería:** Las funciones se agrupan por prefijo
  - ANSI\_: Librería ANSI
  - WIN\_: Librería WINDOW
  - MAT\_: Librería MATRIX
  - CALC\_: Librería CALCULUS
- **Mayúsculas:** Todas las funciones en mayúsculas (aunque BCL es case-insensitive)
- **Guion bajo:** Separador de palabras en nombres compuestos

### 1.5.2. Parámetros

- **Entrada:** Parámetros que la función lee
- **Salida:** Parámetros donde la función escribe resultados (usualmente el último parámetro)
- **Opcionales:** Marcados con [opcional] en la documentación

### 1.5.3. Códigos de ejemplo

Los ejemplos de código en este manual están formateados para claridad:

#### Ejemplo

Ejemplo de código con resultado:

```

1 SET a 5
2 SET b 3
3 SET suma [EXPR $a + $b]
4 PUTS "Suma: $suma"

```

Salida:

Suma: 8

## 1.6. Estructura del documento

Este manual está organizado en los siguientes capítulos:

**Capítulo 2: ANSI** Control de terminal, colores, cursores y caracteres Unicode

**Capítulo 3: WINDOW** Sistema de ventanas en terminal con menús y controles

**Capítulo 4: MATRIX** Operaciones matriciales estilo MATLAB

**Capítulo 5: CALCULUS** Cálculo numérico: derivadas, integrales, métodos numéricos

**Apéndices** Referencia rápida, tablas de funciones y ejemplos completos

## 1.7. Soporte y contribuciones

Para reportar problemas, sugerir mejoras o contribuir al desarrollo de las librerías:

- **GitHub:** <https://github.com/yourusername/bcl>
- **Documentación:** docs/man\_llm.md
- **Ejemplos:** Carpeta examples/

## 1.8. Licencia

Las librerías estándar de BCL se distribuyen bajo la misma licencia que el intérprete BCL. Consulte el archivo LICENSE.txt para más detalles.

# Capítulo 2

## ANSI - Control de Terminal

### 2.1. Introducción

La librería ANSI proporciona control completo del terminal mediante secuencias de escape ANSI y caracteres Unicode para gráficos.

```
1 SOURCE "lib/ANSI.BLB"
```

### 2.2. Control de Pantalla

```
1 ANSI_CLEAR           # Limpiar pantalla
2 ANSI_CURSOR_HOME    # Cursor a (1,1)
3 ANSI_CURSOR_GOTO 10 20 # Mover cursor a fila 10, col 20
4 ANSI_CURSOR_HIDE    # Ocultar cursor
5 ANSI_CURSOR_SHOW    # Mostrar cursor
6 ANSI_RESET           # Reset total de formato
```

### 2.3. Colores

#### 2.3.1. Constantes de Color

**Foreground (texto):**

- ANSI\_FG\_BLACK, ANSI\_FG\_RED, ANSI\_FG\_GREEN, ANSI\_FG\_YELLOW
- ANSI\_FG\_BLUE, ANSI\_FG\_MAGENTA, ANSI\_FG\_CYAN, ANSI\_FG\_WHITE
- Versiones brillantes: ANSI\_FG\_BRIGHT\_\*

**Background (fondo):**

- ANSI\_BG\_\* (mismo esquema)

#### 2.3.2. Uso

```

1 ANSI_SET_COLOR $ANSI_FG_BRIGHT_CYAN $ANSI_BG_BLUE
2 PPUTS "Texto colorido"
3 ANSI_RESET

```

## 2.4. Caracteres Unicode v2.0

### 2.4.1. Box Drawing - Línea Simple

Constante	Descripción	Carácter
ANSI_BOX_TL	Esquina superior izquierda	
ANSI_BOX_TR	Esquina superior derecha	
ANSI_BOX_BL	Esquina inferior izquierda	
ANSI_BOX_BR	Esquina inferior derecha	
ANSI_BOX_H	Línea horizontal	
ANSI_BOX_V	Línea vertical	

### 2.4.2. Box Drawing - Línea Doble

```

1 ANSI_BOX_D_TL      #
2 ANSI_BOX_D_TR      #
3 ANSI_BOX_D_BL      #
4 ANSI_BOX_D_BR      #
5 ANSI_BOX_D_H       #
6 ANSI_BOX_D_V       #

```

### 2.4.3. Box Drawing - Redondeado

```

1 ANSI_BOX_R_TL      #
2 ANSI_BOX_R_TR      #
3 ANSI_BOX_R_BL      #
4 ANSI_BOX_R_BR      #

```

### 2.4.4. Otros Símbolos

```

1 ANSI_ARROW_UP , ANSI_ARROW_DOWN , ANSI_ARROW_LEFT , ANSI_ARROW_RIGHT
2 ANSI_CHECKMARK      #
3 ANSI_CROSSMARK      #
4 ANSI_BULLET          #
5 ANSI_STAR            #
6 ANSI_BLOCK_FULL , ANSI_BLOCK_LIGHT , ANSI_BLOCK_MEDIUM ,
                         ANSI_BLOCK_DARK

```

## 2.5. Ejemplo: Dibujar Caja

```
1 SOURCE "lib/ANSI.BLB"
2
3 ANSI_CLEAR
4 ANSI_CURSOR_GOTO 5 10
5 ANSI_SET_COLOR $ANSI_FG_BRIGHT_CYAN $ANSI_BG_BLACK
6
7 # Línea superior
8 PPUTS -NONEWLINE $ANSI_BOX_TL
9 FOR 0 TO 38 DO
10   PPUTS -NONEWLINE $ANSI_BOX_H
11 END
12 PPUTS $ANSI_BOX_TR
13
14 # Líneas laterales
15 FOR 6 TO 14 DO
16   SET row $__FOR
17   ANSI_CURSOR_GOTO $row 10
18   PPUTS -NONEWLINE $ANSI_BOX_V
19   ANSI_CURSOR_GOTO $row 50
20   PPUTS $ANSI_BOX_V
21 END
22
23 # Línea inferior
24 ANSI_CURSOR_GOTO 15 10
25 PPUTS -NONEWLINE $ANSI_BOX_BL
26 FOR 0 TO 38 DO
27   PPUTS -NONEWLINE $ANSI_BOX_H
28 END
29 PPUTS $ANSI_BOX_BR
30
31 ANSI_RESET
```



## Capítulo 3

# WINDOW - Gestión de Ventanas

### 3.1. Introducción

Sistema completo de ventanas para terminal con soporte para menús, barras de progreso, botones y contenido scrollable.

```
1 SOURCE "lib/WINDOW.BLB"      # Carga automáticamente ANSI.BLB
2 WIN_INIT                      # Inicializar sistema
```

### 3.2. Creación y Dibujo

```
1 WIN_CREATE id row col width height title
2 WIN_DRAW id                         # Dibujar ventana
3 WIN_CLEAR id                        # Limpiar contenido
```

Ejemplo:

```
1 WIN_CREATE 0 5 10 50 15 "Mi Ventana"
2 WIN_SET_BORDER_STYLE 0 1           # 0=simple, 1=doble,
2=redondo
3 WIN_SET_COLOR 0 $ANSI_FG_WHITE $ANSI_BG_BLUE
4 WIN_SET_BORDER_COLOR 0 $ANSI_FG_BRIGHT_CYAN $ANSI_BG_BLUE
5 WIN_DRAW 0
```

### 3.3. Contenido

```
1 WIN_PRINT id row text          # Imprimir en fila
2 WIN_PRINT_CENTER id row text   # Texto centrado
3 WIN_HLINE id row              # Línea horizontal
```

## 3.4. Elementos Interactivos

### 3.4.1. Menú

```
1 WIN_MENU id items selected
```

Ejemplo:

```
1 SET menu "Nuevo | Abrir | Guardar | Salir"
2 WIN_MENU 0 $menu 1 # Item 1 seleccionado
```

### 3.4.2. Barra de Progreso

```
1 WIN_PROGRESS id row percent
2
3 FOR 0 TO 10 DO
4   SET pct [EXPR ${_FOR} * 10]
5   WIN_PROGRESS 0 5 $pct
6 END
```

### 3.4.3. Botones

```
1 WIN_BUTTON id row col label
2
3 WIN_BUTTON 0 10 15 "Aceptar"
4 WIN_BUTTON 0 10 30 "Cancelar"
```

## 3.5. Contenido Scrollable

```
1 WIN_ADD_LINE id text # Añadir línea al buffer
2 WIN_REDRAW_CONTENT id visible # Redibujar contenido
3 WIN_SCROLL id direction # -1=arriba, 1=abajo
```

## 3.6. Gestión de Ventanas

```
1 WIN_HIDE id # Ocultar
2 WIN_SHOW id # Mostrar y redibujar
3 WIN_MOVE id new_row new_col # Mover
4 WIN_MESSAGE_BOX message # Diálogo simple
5 WIN_CLEANUP # Limpiar al salir
```

### 3.7. Ejemplo Completo

```
1 #!/usr/bin/env bcl
2 SOURCE "lib/WINDOW.BLB"
3
4 WIN_INIT
5
6 # Crear ventana principal
7 WIN_CREATE 0 3 5 60 15 "Aplicación"
8 WIN_SET_BORDER_STYLE 0 1
9 WIN_SET_COLOR 0 $ANSI_FG_BRIGHT_WHITE $ANSI_BG_BLUE
10 WIN_DRAW 0
11
12 # Contenido
13 WIN_PRINT_CENTER 0 2 "Bienvenido"
14 WIN_PRINT 0 4 " Seleccione opción:"
15
16 # Menú
17 SET menu "Nuevo|Abrir|Guardar|Salir"
18 WIN_MENU 0 $menu 1
19
20 # Barra de progreso
21 WIN_PRINT 0 10 " Cargando..."
22 WIN_PROGRESS 0 11 75
23
24 # Botones
25 WIN_HLINE 0 12
26 WIN_BUTTON 0 13 15 " OK "
27 WIN_BUTTON 0 13 30 " Cancel "
28
29 # Esperar
30 ANSI_CURSOR_GOTO 24 1
31 PUTS -NONEWLINE "Presione ENTER..."
32 GETS input
33
34 WIN_CLEANUP
```



# Capítulo 4

# MATRIX - Operaciones Matriciales

## 4.1. Introducción

Librería de operaciones matriciales estilo MATLAB para BCL.

### Advertencia

**Importante:** Las matrices deben declararse como GLOBAL en procedimientos.

```
1 PROC MI_FUNCION DO
2   GLOBAL M A B resultado      # Obligatorio
3   MAT_ZEROS M 3 3
4 END
```

```
1 SOURCE "lib/MATRIX.BLB"
```

## 4.2. Creación de Matrices

```
1 MAT_ZEROS name rows cols          # Matriz de ceros
2 MAT_ONES name rows cols           # Matriz de unos
3 MAT_EYE name size                 # Matriz identidad
4 MAT_RAND name rows cols max      # Matriz aleatoria [0,max]
5 MAT_FROM_LIST name rows cols data # Desde lista plana
```

Ejemplo:

```
1 GLOBAL M I A
2
3 MAT_ZEROS M 3 3                  # M = 3x3 de ceros
4 MAT_EYE I 3                      # I = identidad 3x3
5 MAT_FROM_LIST A 2 2 "1 2 3 4"    # A = [[1,2],[3,4]]
```

## 4.3. Operaciones

```

1  MAT_ADD A B C                      #  $C = A + B$ 
2  MAT_SUB A B C                      #  $C = A - B$ 
3  MAT_MUL A B C                      #  $C = A * B$  (multiplicación
   matricial)
4  MAT_SCALAR_MUL A k B              #  $B = k * A$ 
5  MAT_ELEM_MUL A B C                #  $C = A .* B$  (elemento a
   elemento)
6  MAT_TRANSPOSE A A_T               #  $A_T = A'$ 

```

## 4.4. Análisis

```

1  MAT_SUM A total                  # Suma de todos los
   elementos
2  MAT_MEAN A promedio             # Promedio
3  MAT_MIN A minimo               # Valor mínimo
4  MAT_MAX A maximo               # Valor máximo
5  MAT_TRACE A traza              # Traza (suma diagonal)
6  MAT_DET_2X2 A det              # Determinante (solo 2x2)

```

## 4.5. Utilidades

```

1  MAT_PRINT name                  # Imprimir matriz formateada
2  MAT_COPY src dst               # Copiar matriz
3  MAT_FILL name value            # Llenar con valor
4  MAT_GET_ROW name row result   # Extraer fila
5  MAT_GET_COL name col result   # Extraer columna

```

## 4.6. Ejemplo Completo

```

1  #!/usr/bin/env bcl
2  SOURCE "lib/MATRIX.BLB"
3
4  PROC MATRIZ_DEMO DO
5      GLOBAL A B C I resultado
6
7      # Crear matrices
8      MAT_FROM_LIST A 2 2 "1 2 3 4"
9      MAT_FROM_LIST B 2 2 "5 6 7 8"
10     MAT_EYE I 2
11
12     # Operaciones
13     PUTS "A:"
14     MAT_PRINT A
15     PUTS "\nB:"
16     MAT_PRINT B
17
18     MAT_ADD A B C

```

```
19    PUTS "\nA + B:"
20    MAT_PRINT C
21
22    MAT_MUL A B C
23    PUTS "\nA * B:"
24    MAT_PRINT C
25
26    # Análisis
27    MAT_TRACE A tr
28    MAT_DET_2X2 A det
29    PUTS "\nTr(A) = $tr"
30    PUTS "Det(A) = $det"
31 END
32
33 MATRIX_DEMO
```

## 4.7. Almacenamiento

Las matrices se almacenan en el array global `_mat`:

- Metadata: `_mat(name.rows)`, `_mat(name.cols)`
- Elementos: `name(i,j)` donde i=fila, j=columna
- Índices base-0



# Capítulo 5

# CALCULUS - Cálculo Numéricico

## 5.1. Introducción

La librería CALCULUS proporciona un conjunto completo de funciones para cálculo numéricico, incluyendo:

- Derivadas numéricas (diferencias finitas)
- Integración numérica (trapecio, Simpson, punto medio)
- Resolución de ecuaciones diferenciales ordinarias (Euler, Runge-Kutta)
- Búsqueda de raíces (Newton-Raphson, bisección, secante)
- Operaciones con polinomios
- Cálculo de límites y series

### Nota

Esta librería implementa métodos numéricos estándar con precisión configurable. Para aplicaciones que requieren alta precisión, ajuste los parámetros `CALC_EPSILON` y `CALC_MAX_ITER`.

## 5.2. Carga y configuración

```
1 SOURCE "lib/CALCULUS.BLB"
2
3 # Configuración opcional
4 SET CALC_EPSILON 1e-8          # Tolerancia (default: 1e-6)
5 SET CALC_MAX_ITER 5000         # Max iteraciones (default: 1000)
6
7 # Constantes disponibles
8 PUTS $CALC_PI                 #           3.14159...
9 PPUTS $CALC_E                  # e       2.71828...
```

### 5.3. Derivadas Numéricas

#### 5.3.1. CALC\_DERIV\_FORWARD

Calcula la derivada usando diferencias finitas hacia adelante:

$$f'(x) \approx \frac{f(x + h) - f(x)}{h}$$

**Sintaxis:**

```
1 CALC_DERIV_FORWARD expr x h result
```

**Parámetros:**

- **expr:** Expresión de la función (debe usar variable \$x)
- **x:** Punto donde calcular la derivada
- **h:** Paso (típicamente 0.001 a 0.00001)
- **result:** Variable donde almacenar el resultado

**Ejemplo:**

```
1 # Derivada de f(x) = x^2 en x=3
2 # f'(x) = 2x, entonces f'(3) = 6
3 CALC_DERIV_FORWARD "$x * $x" 3.0 0.001 deriv
4 PPUTS "f'(3)      $deriv"
5 # Output: f'(3)      6.001
```

#### 5.3.2. CALC\_DERIV\_CENTRAL

Derivada con diferencias centrales (más precisa):

$$f'(x) \approx \frac{f(x + h) - f(x - h)}{2h}$$

**Ejemplo:**

```
1 # Derivada de sin(x) en x= /2
2 # d/dx[sin(x)] = cos(x), cos( /2) = 0
3 SET x_val [EXPR $CALC_PI / 2.0]
4 CALC_DERIV_CENTRAL "sin($x)" $x_val 0.0001 deriv
5 PPUTS "cos( /2)      $deriv"
6 # Output: cos( /2)      0.000000
```

#### 5.3.3. CALC\_DERIV2

Segunda derivada usando diferencias finitas:

$$f''(x) \approx \frac{f(x + h) - 2f(x) + f(x - h)}{h^2}$$

**Ejemplo:**

```

1 # Segunda derivada de x^3
2 # f(x) = x^3, f''(x) = 6x, f''(2) = 12
3 CALC_DERIV2 "$x * $x * $x" 2.0 0.001 deriv2
4 PPUTS "f''(2)      $deriv2"
5 # Output: f''(2)      12.000

```

## 5.4. Integración Numérica

### 5.4.1. CALC\_INTEGRATE\_TRAP

Integración por regla del trapecio:

$$\int_a^b f(x)dx \approx \frac{h}{2} \left[ f(a) + 2 \sum_{i=1}^{n-1} f(x_i) + f(b) \right]$$

**Sintaxis:**

```
1 CALC_INTEGRATE_TRAP expr a b n result
```

**Parámetros:**

- **expr:** Expresión del integrando
- **a, b:** Límites de integración
- **n:** Número de subintervalos (mayor = más preciso)
- **result:** Variable resultado

**Ejemplo:**

```

1 # Integral de x^2 de 0 a 1
2 #          x   dx = [x /3]      = 1/3      0.333333
3 CALC_INTEGRATE_TRAP "$x * $x" 0 1 100 area
4 PPUTS "Área = $area"
5 # Output: Área = 0.333333

```

### 5.4.2. CALC\_INTEGRATE\_SIMPSON

Regla de Simpson (más precisa que trapecio):

$$\int_a^b f(x)dx \approx \frac{h}{3} \left[ f(a) + 4 \sum f_{odd} + 2 \sum f_{even} + f(b) \right]$$

**Nota:** n debe ser par. Si es impar, se incrementa automáticamente.

**Ejemplo:**

```

1 # Integral de sin(x) de 0 a
2 #          ^    sin(x) dx = [-cos(x)]    ^    = 2
3 CALC_INTEGRATE_SIMPSON "sin($x)" 0 $CALC_PI 100 result
4 PPUTS "Resultado = $result"
5 # Output: Resultado      2.000000

```

### 5.4.3. Comparación de métodos

Método	Precisión	Uso recomendado
Trapecio	$O(h^2)$	Integrales simples, n grande
Simpson	$O(h^4)$	Mayor precisión, funciones suaves
Punto medio	$O(h^2)$	Similar a trapecio

## 5.5. Ecuaciones Diferenciales

### 5.5.1. CALC\_EULER

Método de Euler para resolver EDOs de la forma  $dy/dx = f(x, y)$ .

**Sintaxis:**

```

1 PROC MY_SOLVER DO
2   GLOBAL EULER_X EULER_Y      # Arrays de resultados
3   CALC_EULER expr x0 y0 h n
4 END

```

**Parámetros:**

- **expr:** Expresión  $f(x, y)$  (usa variables \$x e \$y)
- **x0, y0:** Condición inicial  $y(x_0) = y_0$
- **h:** Paso de integración
- **n:** Número de pasos

**Resultado:** Arrays globales EULER\_X(i) y EULER\_Y(i) con  $i=0..n$

**Ejemplo:**

```

1 # Resolver dy/dx = y, y(0) = 1
2 # Solución exacta: y = e^x
3 PROC SOLVE_EXPONENTIAL DO
4   GLOBAL EULER_X EULER_Y
5   CALC_EULER "$y" 0 1 0.1 10
6
7   # Imprimir resultados
8   FOR 0 TO 10 DO
9     SET i $__FOR
10    PUTS "x=$EULER_X($i), y=$EULER_Y($i)"
11  END
12 END
13 SOLVE_EXPONENTIAL

```

### 5.5.2. CALC\_RK4

Método de Runge-Kutta de 4º orden (alta precisión).

Nota

RK4 es significativamente más preciso que Euler para el mismo paso h. Recomendado para problemas que requieren alta exactitud.

**Sintaxis:**

```

1 PROC MY_SOLVER DO
2   GLOBAL RK4_X RK4_Y
3   CALC_RK4 expr x0 y0 h n
4 END

```

**Ejemplo:**

```

1 # Resolver dy/dx = -2xy, y(0) = 1
2 # (Decaimiento Gaussiano)
3 PROC SOLVE_GAUSSIAN DO
4   GLOBAL RK4_X RK4_Y
5   CALC_RK4 "-2 * $x * $y" 0 1 0.1 20
6
7   # Valor final
8   PUTS "y(2.0)      $RK4_Y(20)"
9 END
10 SOLVE_GAUSSIAN

```

## 5.6. Búsqueda de Raíces

### 5.6.1. CALC\_NEWTON

Método de Newton-Raphson para encontrar raíces.

**Fórmula:**

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

**Sintaxis:**

```

1 CALC_NEWTON f_expr df_expr x0 tol max_iter result

```

**Parámetros:**

- **f\_expr:** Expresión de  $f(x)$
- **df\_expr:** Expresión de  $f'(x)$
- **x0:** Estimación inicial
- **tol:** Tolerancia (ej. 1e-6)
- **max\_iter:** Máximo de iteraciones
- **result:** Variable resultado

**Ejemplo:**

```

1 # Encontrar 2 resolviendo x^2 - 2 = 0
2 CALC_NEWTON "$x*$x - 2" "2*$x" 1.0 1e-8 100 root
3 PUTS " 2      $root"
4 # Output: 2      1.41421356

```

### 5.6.2. CALC\_BISECTION

Método de la bisección (más robusto, convergencia garantizada).

**Requisito:**  $f(a) \cdot f(b) < 0$  (signos opuestos).

**Sintaxis:**

```
1 CALC_BISECTION expr a b tol max_iter result
```

**Ejemplo:**

```
1 # Encontrar raíz de cos(x) en [0, /2]
2 CALC_BISECTION "cos($x)" 0 [EXPR $CALC_PI/2] 1e-6 100 root
3 PUTS "Raíz      $root"
4 # Output: Raíz      1.570796 ( /2)
```

### 5.6.3. CALC\_SECANT

Método de la secante (no requiere derivada).

**Fórmula:**

$$x_{n+1} = x_n - f(x_n) \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$

**Sintaxis:**

```
1 CALC_SECANT expr x0 x1 tol max_iter result
```

**Ejemplo:**

```
1 # Encontrar raíz de x^3 - x - 1 = 0
2 CALC_SECANT "$x*$x*$x - $x - 1" 1.0 2.0 1e-6 100 root
3 PUTS "Raíz      $root"
4 # Output: Raíz      1.324718
```

### 5.6.4. Comparación de métodos

Método	Convergencia	Derivada	Robustez
Newton	Cuadrática	Requiere	Media
Bisección	Lineal	No requiere	Alta
Secante	Superlineal	No requiere	Media

## 5.7. Polinomios

### 5.7.1. CALC\_POLY\_EVAL

Evalúa polinomio usando el método de Horner.

**Sintaxis:**

```
1 PROC EVAL_POLYNOMIAL DO
2   GLOBAL coeffs
3   SET coeffs(0) 1      # a
4   SET coeffs(1) 2      # a
5   SET coeffs(2) 3      # a
```

```

6      # P(x) = 1 + 2x + 3x
7
8      CALC_POLY_EVAL 2 5.0 result
9      PUTS "P(5) = $result"
10     # P(5) = 1 + 2(5) + 3(25) = 86
11 END

```

### 5.7.2. CALC\_POLY\_ROOTS\_QUAD

Encuentra raíces de ecuación cuadrática  $ax^2 + bx + c = 0$ .

**Sintaxis:**

```

1 PROC FIND_ROOTS DO
2   GLOBAL POLY_ROOT1 POLY_ROOT2 POLY_DISCRIMINANT
3   CALC_POLY_ROOTS_QUAD a b c
4 END

```

**Ejemplo:**

```

1 # Resolver x^2 - 5x + 6 = 0
2 # Raíces: x = 2, x = 3
3 PROC SOLVE_QUADRATIC DO
4   GLOBAL POLY_ROOT1 POLY_ROOT2 POLY_DISCRIMINANT
5   CALC_POLY_ROOTS_QUAD 1 -5 6
6   PUTS " x = $POLY_ROOT1"
7   PUTS " x = $POLY_ROOT2"
8 END
9 SOLVE_QUADRATIC

```

## 5.8. Series y Límites

### 5.8.1. CALC\_SERIES\_SUM

Calcula suma de serie  $\sum_{n=start}^{end} f(n)$ .

**Ejemplo:**

```

1 # Aproximar /6 con serie: (1/n) n=1 to
2 CALC_SERIES_SUM "1.0 / ($n * $n)" 1 10000 sum
3 SET pi_sq_6 [EXPR $sum]
4 SET pi_approx [EXPR sqrt($pi_sq_6 * 6.0)]
5 PUTS "$pi_approx"

```

### 5.8.2. CALC\_LIMIT

Estimación numérica de límites.

**Ejemplo:**

```

1 # lim( x 0 ) sin(x)/x = 1
2 CALC_LIMIT "sin($x) / $x" 0 limit
3 PUTS "límite = $limit"
4 # Output: límite 1.000000

```

## 5.9. Funciones Utilidad

### 5.9.1. CALC\_FACTORIAL

```

1 CALC_FACTORIAL 5 result
2 PUTS "5! = $result"
3 # Output: 5! = 120

```

### 5.9.2. CALC\_COMBINATION

Calcula combinaciones  $C(n, k) = \frac{n!}{k!(n-k)!}$ .

```

1 CALC_COMBINATION 10 3 result
2 PPUTS "C(10,3) = $result"
3 # Output: C(10,3) = 120

```

### 5.9.3. CALC\_GCD

Máximo común divisor (algoritmo de Euclides).

```

1 CALC_GCD 48 18 result
2 PPUTS "GCD(48,18) = $result"
3 # Output: GCD(48,18) = 6

```

## 5.10. Ejemplo Completo: Análisis de Función

```

1 #!/usr/bin/env bcl
2 SOURCE "lib/CALCULUS.BLB"
3
4 # Analizar f(x) = x^3 - 3x^2 + 2x
5 PPUTS "Análisis de f(x) = x^3 - 3x^2 + 2x"
6 PPUTS "===== "
7
8 # 1. Encontrar derivada en x=1
9 CALC_DERIV_CENTRAL "$x*$x*$x - 3*$x*$x + 2*$x" 1.0 0.0001 deriv
10 PPUTS "f'(1) = $deriv"
11
12 # 2. Calcular segunda derivada en x=1
13 CALC_DERIV2 "$x*$x*$x - 3*$x*$x + 2*$x" 1.0 0.0001 deriv2
14 PPUTS "f''(1) = $deriv2"
15
16 # 3. Encontrar raíz cerca de x=2
17 CALC_NEWTON "$x*$x*$x - 3*$x*$x + 2*$x" \
18             "3*$x*$x - 6*$x + 2" 2.0 1e-8 100 root
19 PPUTS "Raíz encontrada: x = $root"
20
21 # 4. Calcular área bajo la curva [0,1]
22 CALC_INTEGRATE_SIMPSON "$x*$x*$x - 3*$x*$x + 2*$x" \
23                           0 1 100 area
24 PPUTS "Área [0,1] = $area"

```

```
25  
26 PUTS "=====
```

## 5.11. Notas de Precisión

### Advertencia

Los métodos numéricos tienen limitaciones inherentes:

- **Error de redondeo:** Acumulado en operaciones flotantes
- **Error de truncamiento:** Por aproximación de derivadas/integrales
- **Convergencia:** No garantizada para todos los problemas

Recomendaciones:

- Usar pasos pequeños ( $h$ ) para derivadas/integrales
- Verificar convergencia en métodos iterativos
- Comparar resultados con diferentes parámetros
- Para alta precisión, considerar librerías especializadas



# Apéndice A

## Referencia Rápida

### A.1. Resumen de Librerías

Librería	Archivo	Funciones
ANSI	ANSI.BLB	Control terminal, colores, Unicode
WINDOW	WINDOW.BLB	Ventanas, menús, widgets
MATRIX	MATRIX.BLB	Operaciones matriciales
CALCULUS	CALCULUS.BLB	Cálculo numérico

### A.2. Tabla de Funciones por Librería

#### A.2.1. ANSI (21 constantes + funciones)

**Control:** CLEAR, CURSOR\_HOME, CURSOR\_GOTO, CURSOR\_HIDE, CURSOR\_SHOW, RESET

**Colores:** 16 FG + 16 BG, SET\_COLOR, SET\_STYLE

**Unicode:** 60+ caracteres (box drawing, flechas, símbolos)

#### A.2.2. WINDOW (26 funciones)

CREATE, DRAW, PRINT, PRINT\_CENTER, SET\_COLOR, SET\_BORDER\_STYLE, SET\_BORDER\_COLOR, CLEAR, HIDE, SHOW, MOVE, MENU, PROGRESS, BUTTON, HLINE, MESSAGE\_BOX, ADD\_LINE, REDRAW\_CONTENT, SCROLL, CLEA-NUP, INIT

#### A.2.3. MATRIX (23 funciones)

CREATE, ZEROS, ONES, EYE, RAND, FROM\_LIST, PRINT, ADD, SUB, MUL, SCALAR\_MUL, ELEM\_MUL, TRANSPOSE, SUM, MEAN, MIN, MAX, TRACE, DET\_2X2, COPY, FILL, GET\_ROW, GET\_COL

#### A.2.4. CALCULUS (27 funciones)

**Derivadas:** DERIV\_FORWARD, DERIV\_BACKWARD, DERIV\_CENTRAL, DERIV2

**Integración:** INTEGRATE\_TRAP, INTEGRATE\_SIMPSON, INTEGRATE\_MIDPOINT

**EDOs:** EULER, RK2, RK4

**Raíces:** NEWTON, BISECTION, SECANT

**Polinomios:** POLY\_EVAL, POLY\_ROOTS\_QUAD

**Series:** LIMIT, SERIES\_SUM

**Utilidades:** FACTORIAL, COMBINATION, GCD

### A.3. Caracteres Unicode Box Drawing

Estilo	Horizontal	Vertical	Esquinas
Simple			
Doble			
Redondeado			

### A.4. Valores Predeterminados

```
1 # CALCULUS
2 CALC_EPSILON = 1e-6
3 CALC_MAX_ITER = 1000
4 CALC_PI = 3.14159265358979323846
5 CALC_E = 2.71828182845904523536
6
7 # WINDOW
8 WIN_COUNT = 0 (incremental)
```

## Apéndice B

# Ejemplos Adicionales

### B.1. Gráfico de Función con Ventanas

```
1 #!/usr/bin/env bcl
2 SOURCE "lib/WINDOW.BLB"
3 SOURCE "lib/CALCULUS.BLB"
4
5 WIN_INIT
6
7 # Ventana para gráfico
8 WIN_CREATE 0 2 2 76 22 "Gráfico de f(x) = sin(x)"
9 WIN_SET_BORDER_STYLE 0 1
10 WIN_DRAW 0
11
12 # Generar puntos
13 PROC GENERATE_PLOT DO
14     GLOBAL RK4_X RK4_Y
15     # Usar RK4 para resolver dy/dx = cos(x), y(0)=0
16     # Solución: y = sin(x)
17     CALC_RK4 "cos($x)" 0 0 0.1 63
18
19     FOR 0 TO 62 DO
20         SET i $__FOR
21         SET x $RK4_X($i)
22         SET y $RK4_Y($i)
23
24         # Escalar a coordenadas de ventana
25         SET row [EXPR 11 - int($y * 10)]
26         SET col [EXPR 2 + $i]
27
28         IF [EXPR $row >= 1 && $row <= 20] THEN
29             ANSI_CURSOR_GOTO [EXPR $row + 2] [EXPR $col + 2]
30             PUTS -NONEWLINE "*"
31         END
32     END
33 END
34
35 GENERATE_PLOT
36
37 ANSI_CURSOR_GOTO 24 1
```

```

38 PUTS -NONEWLINE "Presione ENTER . . ."
39 GETS input
40
41 WIN_CLEANUP

```

## B.2. Calculadora Matricial Interactiva

```

1 #!/usr/bin/env bcl
2 SOURCE "lib/WINDOW.BLB"
3 SOURCE "lib/MATRIX.BLB"
4
5 PROC CALC_MATRIX_INTERACTIVE DO
6     GLOBAL M1 M2 result
7
8     WIN_INIT
9
10    # Ventana principal
11    WIN_CREATE 0 2 5 70 20 "Calculadora Matricial"
12    WIN_SET_BORDER_STYLE 0 1
13    WIN_DRAW 0
14
15    # Crear matrices de ejemplo
16    MAT_FROM_LIST M1 2 2 "1 2 3 4"
17    MAT_FROM_LIST M2 2 2 "5 6 7 8"
18
19    WIN_PRINT 0 1 "Matriz A:"
20    WIN_PRINT 0 2 " [1 2]"
21    WIN_PRINT 0 3 " [3 4]"
22
23    WIN_PRINT 0 5 "Matriz B:"
24    WIN_PRINT 0 6 " [5 6]"
25    WIN_PRINT 0 7 " [7 8]"
26
27    # Menú de operaciones
28    WIN_PRINT 0 9 "Operación:"
29    SET ops "A+B|A-B|A*B|Det(A)|Salir"
30    WIN_MENU 0 $ops 1
31
32    # Realizar operación (simplificado - selección fija)
33    MAT_ADD M1 M2 result
34    WIN_PRINT 0 14 "Resultado (A+B):"
35    WIN_PRINT 0 15 " [6 8]"
36    WIN_PRINT 0 16 " [10 12]"
37
38    WIN_HLINE 0 17
39    WIN_BUTTON 0 18 25 " Calcular "
40
41    ANSI_CURSOR_GOTO 24 1
42    PUTS -NONEWLINE "Presione ENTER . . ."
43    GETS input
44
45    WIN_CLEANUP
46 END

```

47  
48    CALC\_MATRIX\_INTERACTIVE



# Apéndice C

## Solución de Problemas

### C.1. Errores Comunes

#### C.1.1. Arrays no persisten en procedimientos

**Error:** "no such variable"

**Solución:** Declarar arrays como GLOBAL

```
1 PROC MAL DO
2     MAT_ZEROS M 3 3      # ERROR
3 END
4
5 PROC BIEN DO
6     GLOBAL M           # CORRECTO
7     MAT_ZEROS M 3 3
8 END
```

#### C.1.2. Terminal no muestra Unicode

**Síntoma:** Caracteres extraños en lugar de box drawing

**Solución:**

1. Verificar terminal soporta UTF-8
2. Configurar locale: `export LANG=es_ES.UTF-8`
3. Usar terminal moderna (xterm, gnome-terminal, etc.)

#### C.1.3. Métodos numéricos no convergen

**Solución:**

- Verificar condiciones iniciales
- Ajustar tolerancia (`CALC_EPSILON`)
- Aumentar iteraciones máximas
- Verificar que la función tiene raíz/solución

## C.2. Optimización

### C.2.1. Performance de Matrices

Para matrices grandes, considerar:

- Usar operaciones elemento a elemento cuando sea posible
- Evitar copias innecesarias
- BCL no está optimizado para matrices masivas ( $> 100 \times 100$ )

### C.2.2. Integración Numérica

- Simpson es más preciso que trapecio
- Para misma precisión, Simpson requiere menos puntos
- Elegir  $n$  según precisión deseada

## C.3. Recursos Adicionales

- **Documentación principal:** `docs/man_llm.md`
- **Ejemplos:** Carpeta `examples/`
- **Tests:** Carpeta `tests/`
- **Código fuente:** `lib/*.BLB`

**Fin del Manual de Librerías BCL v2.0**

Para más información, visite:

<https://github.com/yourusername/bcl>