

Utilitarios

Parte I

Enrique E. Tarifa, Facultad de Ingeniería, UNJu

Paradigmas

Paradigmas de programación

Programación imperativa

- Detalla los pasos para resolver un problema.
- Fortran, Java, C, Python, Julia, Ruby

```
// C
int total = 0;
for(int i = 0; i < 10; i++){
    total++;
    printf("%i\n", total);
}
```

Programación declarativa

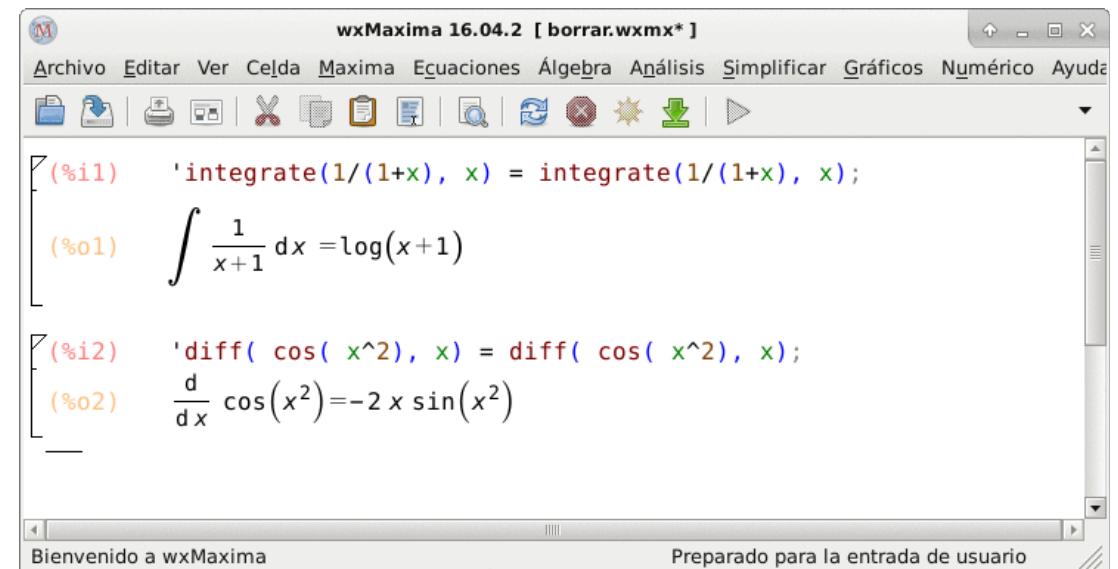
- Expresar el problema a resolver, sin dar los pasos.
- LISP, Haskell, Prolog, SQL, Elixir, XPath

```
-- Haskell
sum[1..10]
```

Aplicaciones declarativas

wxMaxima

- Cálculo simbólico
 - Integración
 - Diferenciación
 - ODEs
 - AEs
 - Transformada de Fourier
- Solución analítica





Letras Griegas ✖

α	β	γ	δ	ε
ζ	η	θ	ι	κ
λ	ν	ξ	π	ρ
σ	τ	υ	φ	χ
ψ	ω			
Γ	Δ	Θ	Λ	Ξ
Π	Σ	Φ	Ψ	Ω

Matemática Común ✖

- Simplificar
- Simplificar (r)
- Factorizar
- Expandir
- Forma Cartesiana
- Sust...
- Canónico (tr)
- Simplificar (tr)

Símbolos Matem... ✖

$\frac{1}{2}$	2	3	$\sqrt{}$
i	e	\hbar	∞
\exists	\nexists	\Rightarrow	∞
\mathbb{Q}	\mathbb{R}	\mathbb{C}	$/$
\mathbb{N}	\mathbb{Z}	\mathbb{Q}	\mathbb{C}
\pm	\mp	\sqcup	\sqcap
\sqsubseteq	\sqsubset	\sqsupset	\sqsupseteq
∇	\int	\approx	\propto

1 Resolución de ODEs

(%i12) $\text{eqn_1: 'diff(f(x),x)='diff(g(x),x)+sin(x);}$
 $\text{eqn_2: 'diff(g(x),x,2)='diff(f(x),x)-cos(x);}$

$$(\text{eqn_1}) \frac{d}{dx} f(x) = \frac{d}{dx} g(x) + \sin(x)$$

$$(\text{eqn_2}) \frac{d^2}{dx^2} g(x) = \frac{d}{dx} f(x) - \cos(x)$$

(%i14) $\text{atvalue('diff(g(x),x),x=0,a)\$}$
 $\text{atvalue(f(x),x=0,1)\$}$

(%i15) $\text{desolve([\text{eqn_1}, \text{eqn_2}], [f(x),g(x)]);}$

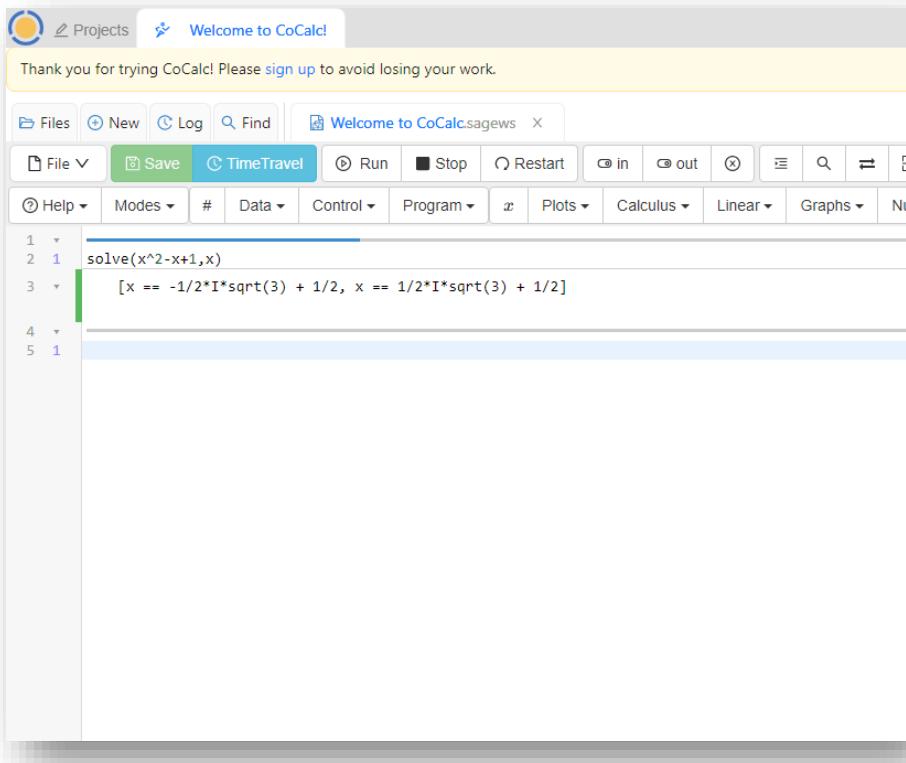
$$(\%o15) \left[f(x) = a \%e^x - a + 1, g(x) = \cos(x) + a \%e^x - a + g(0) - 1 \right]$$

Índice de Contenidos ✖

Resolución de ODEs

Otros CAS (computer algebra system)

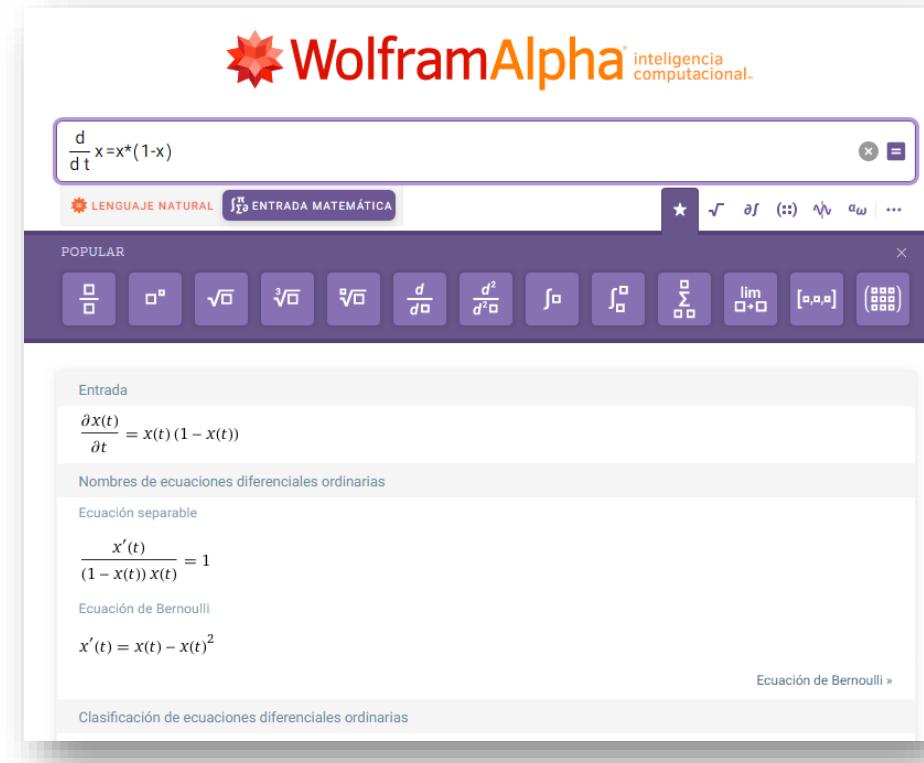
SageMath



The screenshot shows the SageMath interface within CoCalc. The top bar includes 'Projects', 'Welcome to CoCalc!', 'File', 'New', 'Log', 'Find', and a tab for 'Welcome to CoCalc.sagews'. The main workspace displays the following code:

```
1 solve(x^2-x+1,x)
2 [x == -1/2*I*sqrt(3) + 1/2, x == 1/2*I*sqrt(3) + 1/2]
```

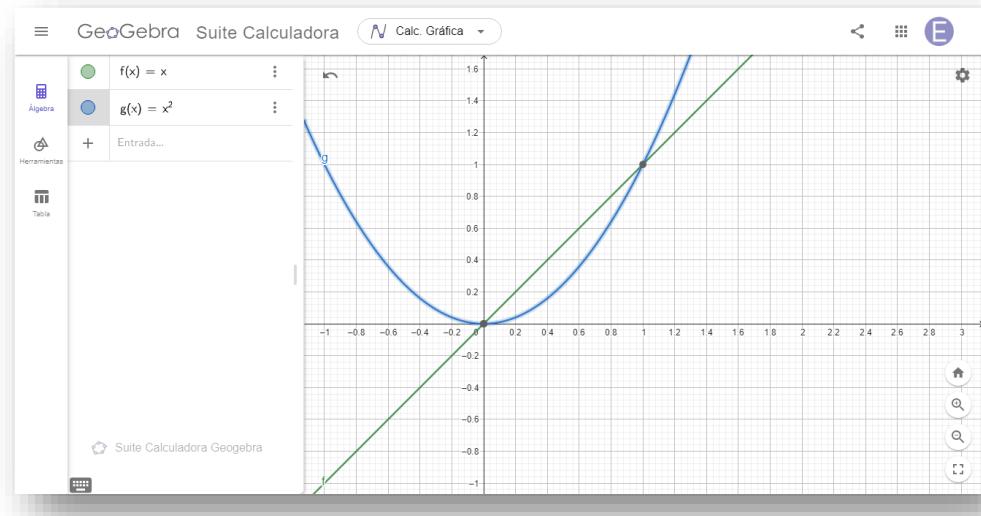
WolframAlpha



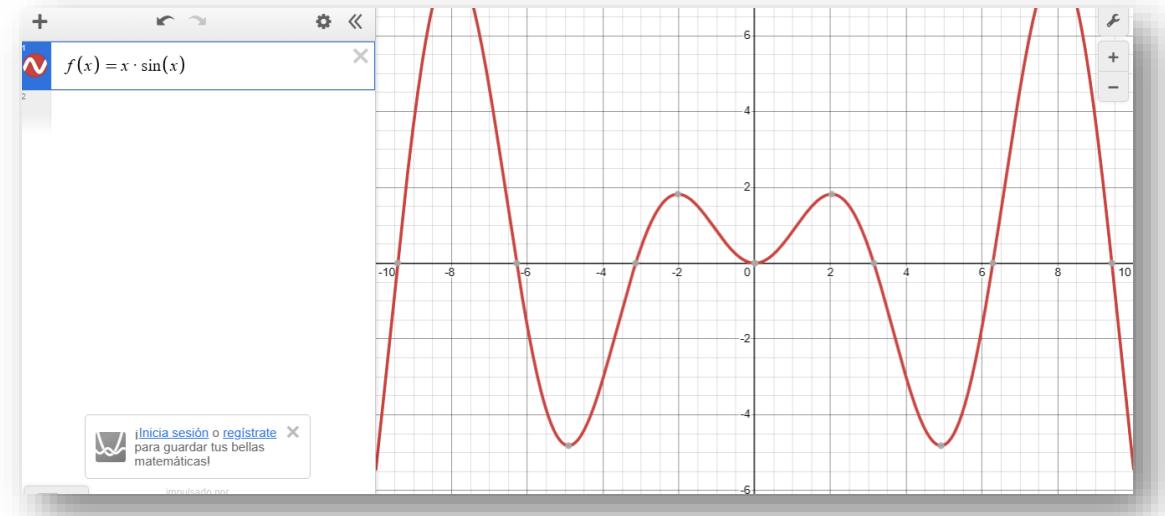
The screenshot shows the WolframAlpha interface. The top bar features the WolframAlpha logo and the text 'inteligencia computacional.'. Below the bar, a search bar contains the differential equation $\frac{dx}{dt} = x(t)(1-x(t))$. The interface includes a 'LENGUAJE NATURAL' button and a 'ENTRADA MATEMÁTICA' button. A toolbar below the search bar contains various mathematical operators like square root, derivative, integral, and summation. The main area displays the equation as 'Ecuación separable' and provides the solution $x'(t) = x(t)(1-x(t))$.

Graficadores

Geogebra

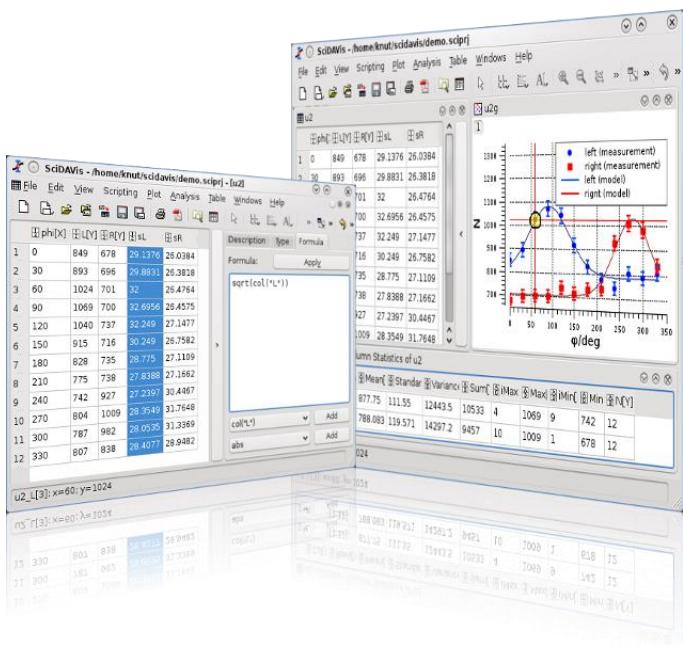


Desmos

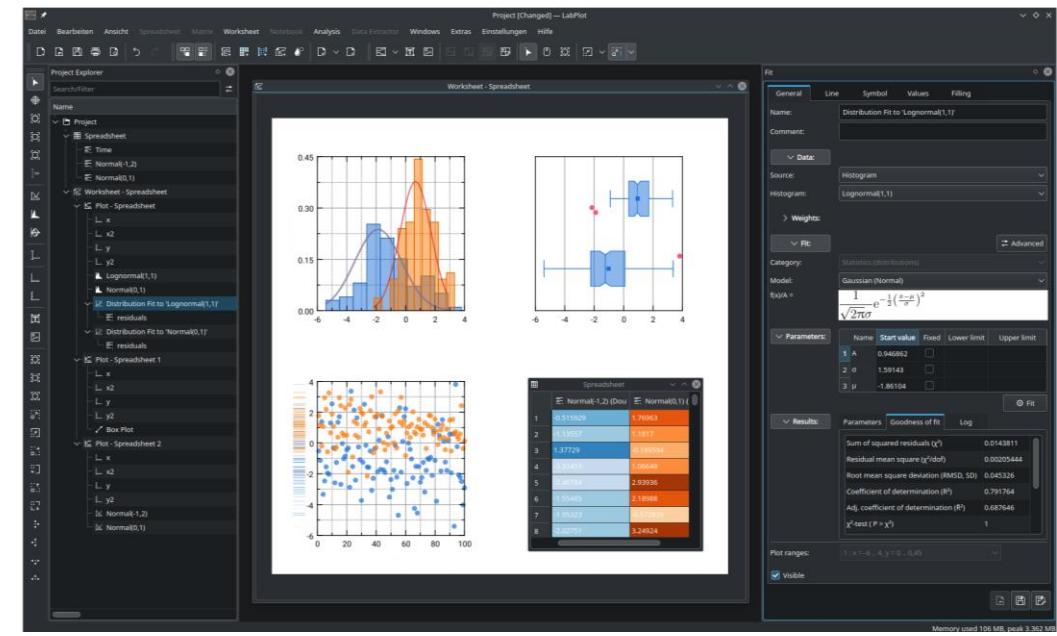


Graficadores

SciDavis



LabPlot



CalcMe

CalcMe  Sistema de ecuaciones Hoja 1 + Archivo ? ☰

Símbolos Aritmética Polinomios Estadística Funciones Cálculo Álgebra lineal Combinatoria Lógica y conjuntos Resolver Griego Unidades de medida Gráficas Programación

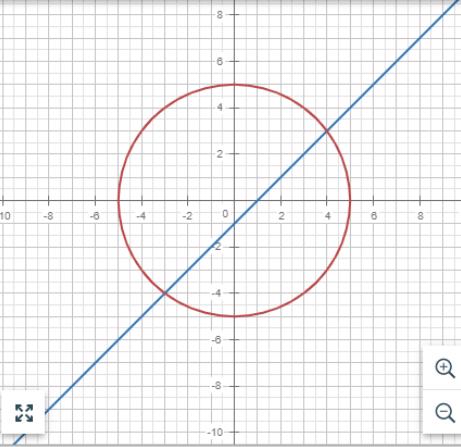
Ejemplo obtenido de <https://youtube.com/watch?v=R25XlyQKJYg&feature=shares&t=434>

$$\begin{aligned}x - y &= 1 &\longrightarrow x = -3 \vee x = 4 &\text{ Solucionar} \\x^2 + y^2 &= 25 &y = -4 & y = 3\end{aligned}$$

$x - y = 1$  Dibujar: tablero1

$x^2 + y^2 = 25$  Dibujar: tablero1

+  ×



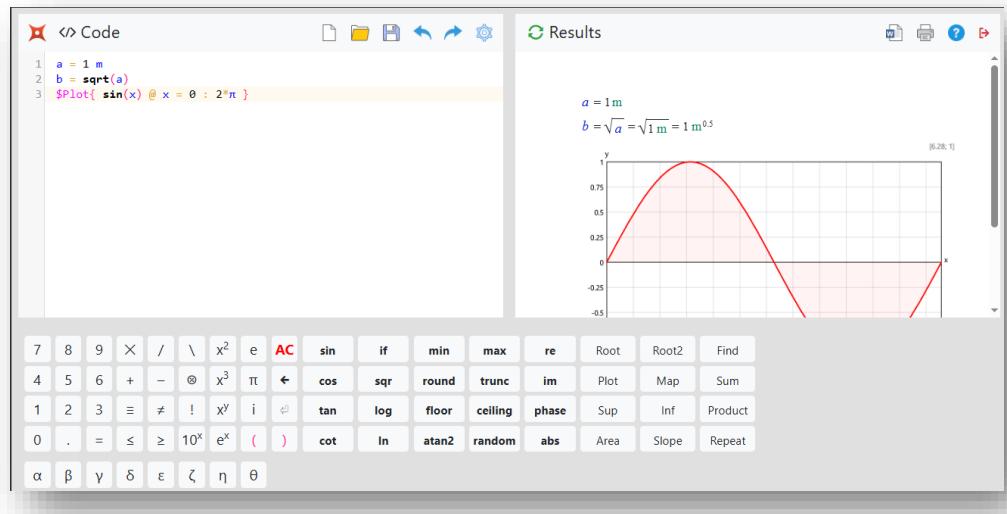
=

Cookie Preferences

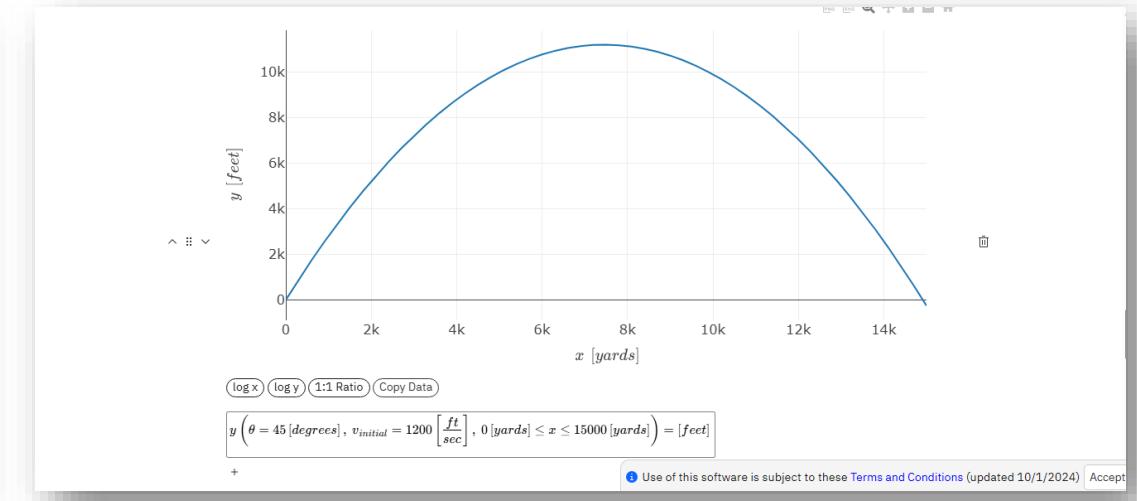
CalcMe

Calculadoras

CalcPad



EngineeringPaper.xyz



ISIM ESL

ISIM International Simulation Limited

Home About Us Products Documents Consultancy Downloads Help Pages Try ESL Online

Home > Try ESL Online > Compile/Run ESL

Esl Examples ▾

ESL Text x

```
-- Tanque
--
STUDY
--
MODEL TANQUE;
-----
-- Tanque con descarga gravitatoria
-----
REAL: L, F;
CONSTANT REAL: F0/20E-3/, A/0.785/, Cv/4.039E-4/, rho/1000.0/, g/9.81/, x/0.25/;
INITIAL
L:= 1.0;

DYNAMIC
-- Calcula el caudal de descarga
F := Cv*x*sqrt(rho*g*L);

-- Calcula el nivel
L := F/A;
```

Output x

```
**** E S L Compiler (Lite) v8.2.5.9
**** Copyright (C) ISIM International Simulation Limited 1992-2022.
< TANQUE      0  WARNINGS      0  ERRORS >
< EXP$MN      0  WARNINGS      0  ERRORS >
Compilation succeeded

**** E S L Interpreter Run-time v8.2.5.9
**** Copyright (C) ISIM International Simulation Limited 1992-2022.
Ejemplo del tanque
Run finished
```

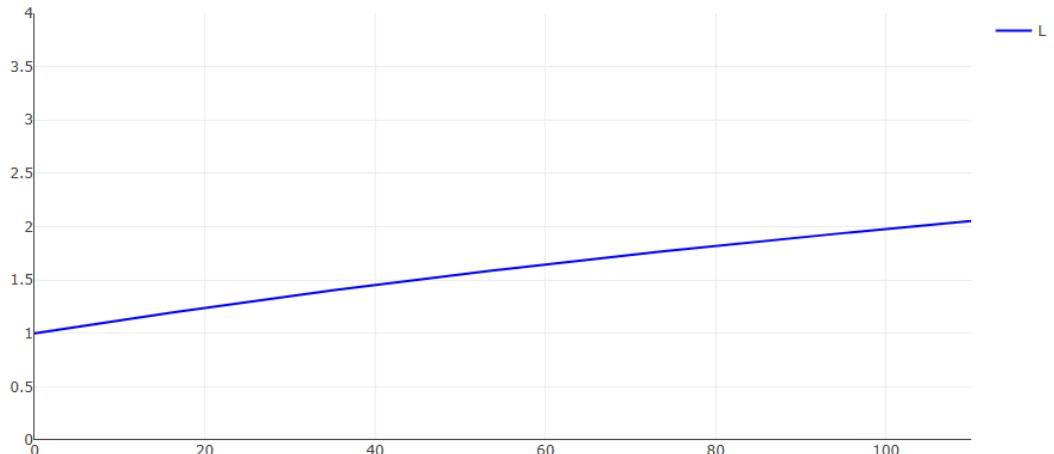
Copyright © ISIM International Simulation Ltd. 2018

ISIM ESL

ISIM ESL

Plot plot 1

TANQUE Plot 1/2



Copyright © ISIM International Simulation Ltd. 2018

```
-- Tanque
--
STUDY
--
MODEL TANQUE;
-----
-- Tanque con descarga gravitatoria
-----
REAL: L, F;
CONSTANT REAL: F0/20E-3/, A/0.785/, Cv/4.039E-4/, rho/1000.0/, g/9.81/, x/0.25/;

INITIAL
  L := 1.0;

DYNAMIC
-- Calcula el caudal de descarga
  F := Cv*x*sqrt(rho*g*L);

-- Calcula el nivel
  L' := (F0-F)/A;

STEP
  PLOT "TANQUE Plot 1/2", t, L, 0, TFIN,0,4;
  PLOT "TANQUE Plot 2/2", t, F, 0, TFIN,0,2*F0;
END TANQUE;
-----
--EXPERIMENT
PRINT "Ejemplo del tanque";

TFIN := 110.0;
CINT := 1;
TANQUE;

END_STUDY
```

PolymathPlus

- Aplicación online
- Resolución de varios tipos de problemas independientes:
 - ODEs
 - AEs
 - Lineales
 - No lineales
 - Regresión

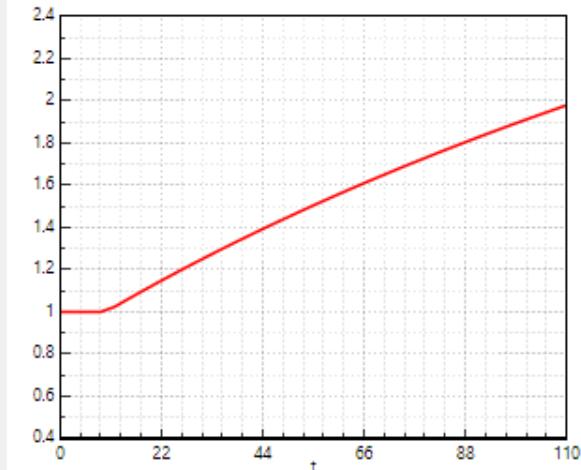
```
1 # Tanque con descarga gravitatoria
2
3 # Inicialización
4 t|0:110
5 L|1
6
7 # ODEs
8 L' = (F0-F)/A
9
10 # AEs
11 F = Cv*x*sqrt(rho*g*L)
12
13 # Parámetros
14 F0 = 20E-3
15 A = 0.785
16 Cv = 4.039E-4
17 rho = 1000
18 g = 9.81
19 x = 0.5
```

```

2
3 # Inicialización
4 t|0:110
5 L|1
6
7 # ODEs
8 L' = (F0-F)/A
9
10 # AEs
11 F = Cv*x*sqrt(rho*g*L)
12
13 # Parámetros
14 F0 = 20E-3
15 A = 0.785
16 Cv = 4.039E-4
17 rho = 1000
18 g = 9.81
19 x = if t < 10 then 0.5 else 0.25
20

```

Integration chart



Formatted equations

$$\frac{d(L)}{d(t)} = \frac{F_0 - F}{A}$$

$$F_0 = 20E-3$$

$$A = 0.785$$

$$Cv = 4.039E-4$$

$$\rho = 1000$$

$$g = 9.81$$

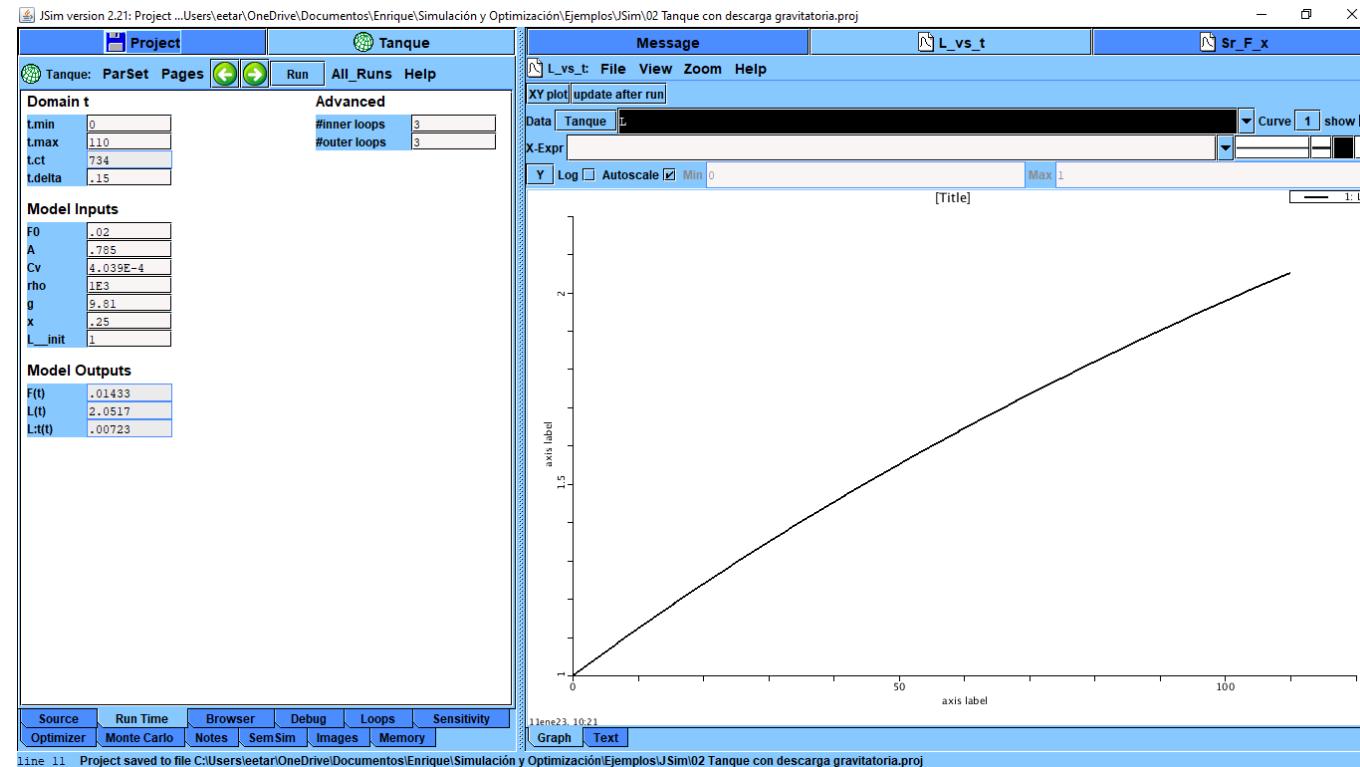
$$F = Cv \cdot x \cdot \sqrt{\rho \cdot g \cdot L}$$

Differential equations

$$1 \frac{d(L)}{d(t)} = \frac{(F_0 - F)}{A}$$

Explicit equations

JSim



JSim

JSim

- Declarativo
- Unidades físicas
- Límites de variables
- Vectores
- Eventos
- Resolución sistemas AEs
- Integración de ODEs
- Integración de PDEs
- Optimización
- Simulación de Monte Carlo

```
// Tanque con descarga gravitatoria

math tanque {

    // Parámetros de simulación
    realDomain t; // tiempo
    t.min     = 0;
    t.max     = 110;
    t.delta   = 0.15;

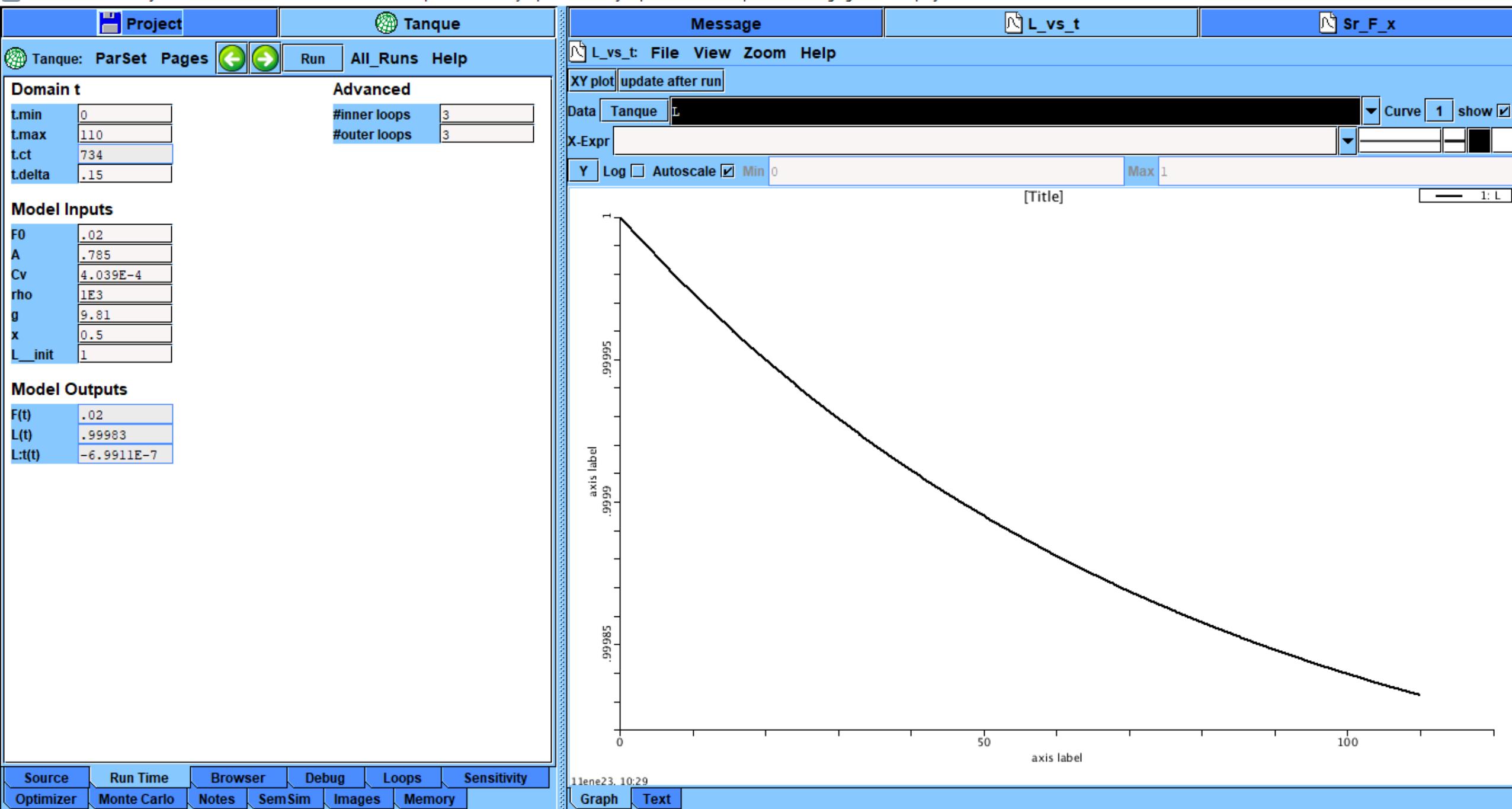
    // Parámetros
    real F0 = 20E-3, A = 0.785, Cv = 4.039E-4,
          rho = 1000, g = 9.81, x = 0.5;

    // Variables
    real F(t);
    real L(t); // Variable de estado

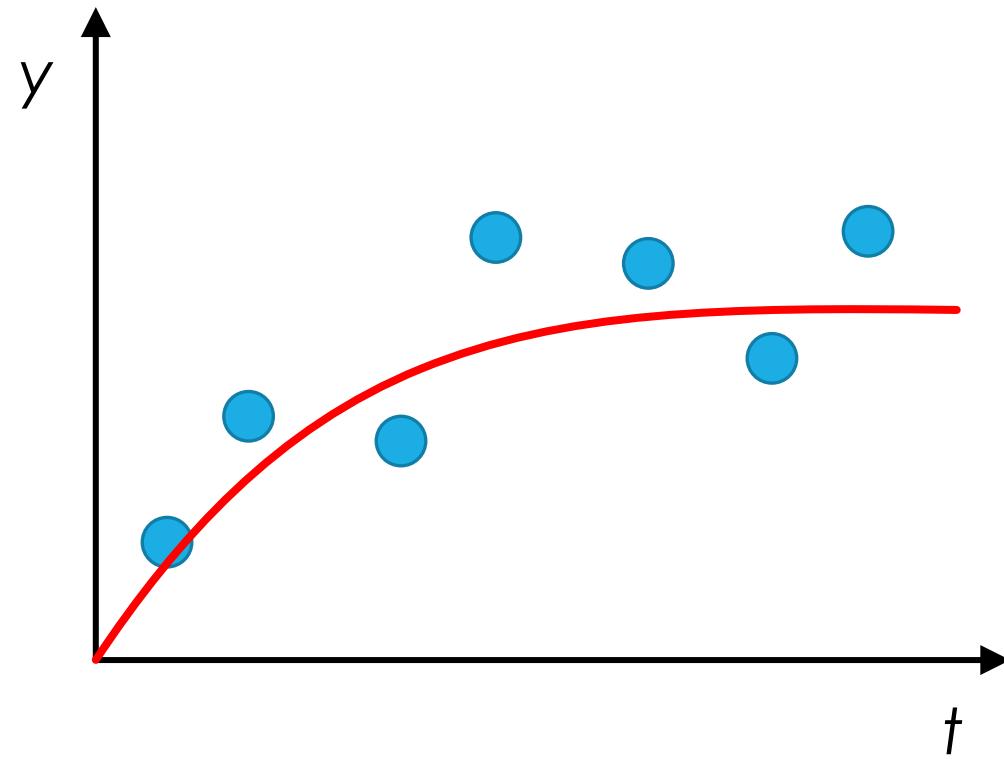
    // Condiciones iniciales
    when(t=t.min) {
        L = 1;
    }

    // ODES
    L:t = (F0-F)/A;

    // AEs
    F - Cv*x*sqrt(rho*g*L) = 0;
}
```

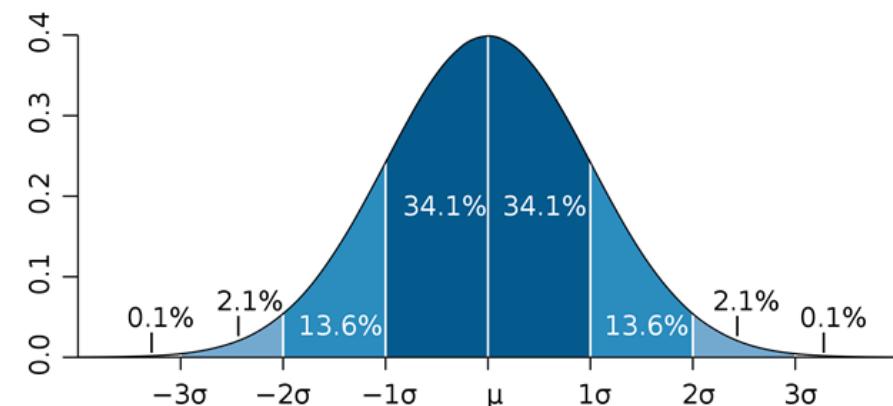


Regresión

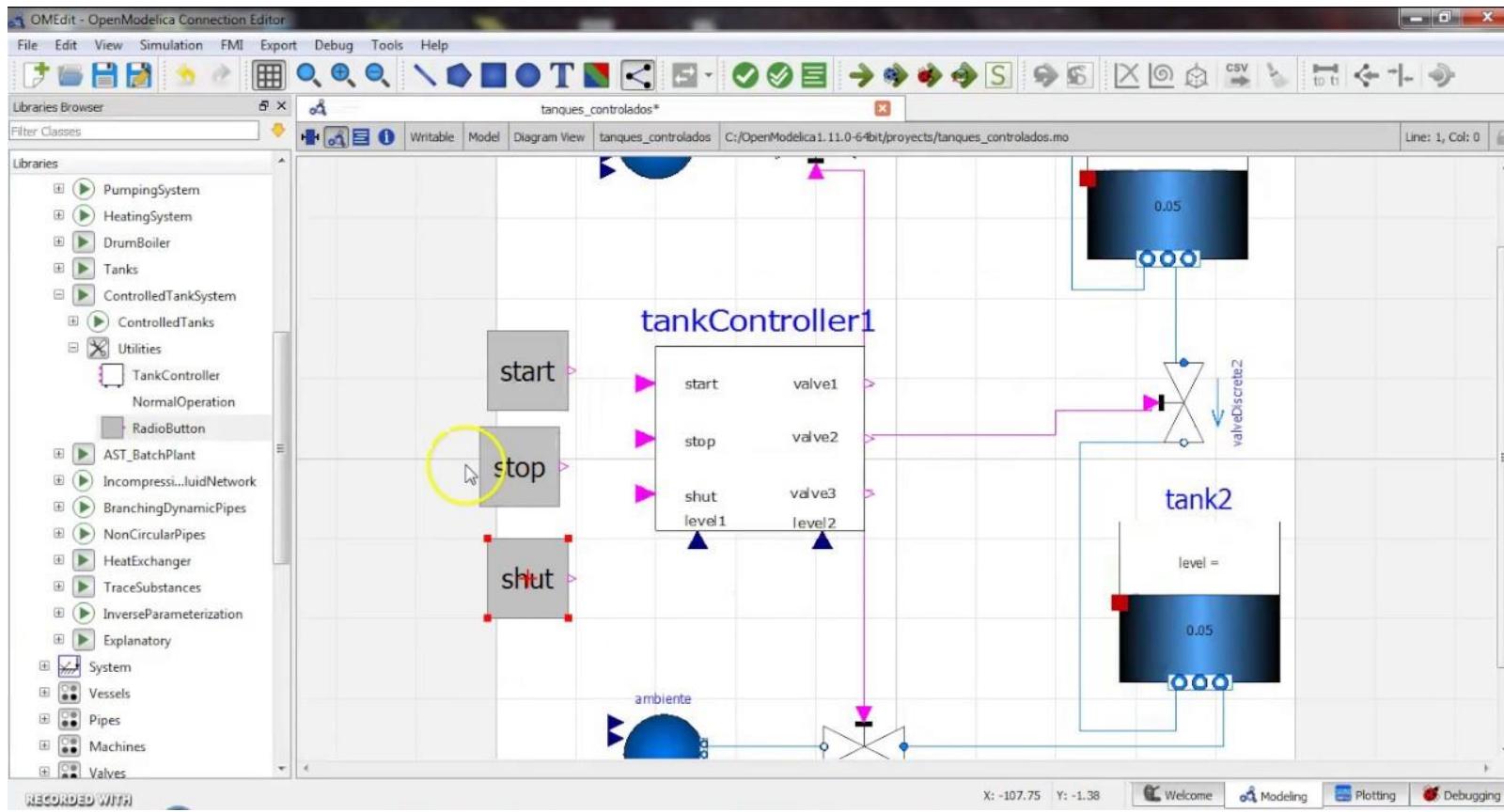


Simulación de Monte Carlo

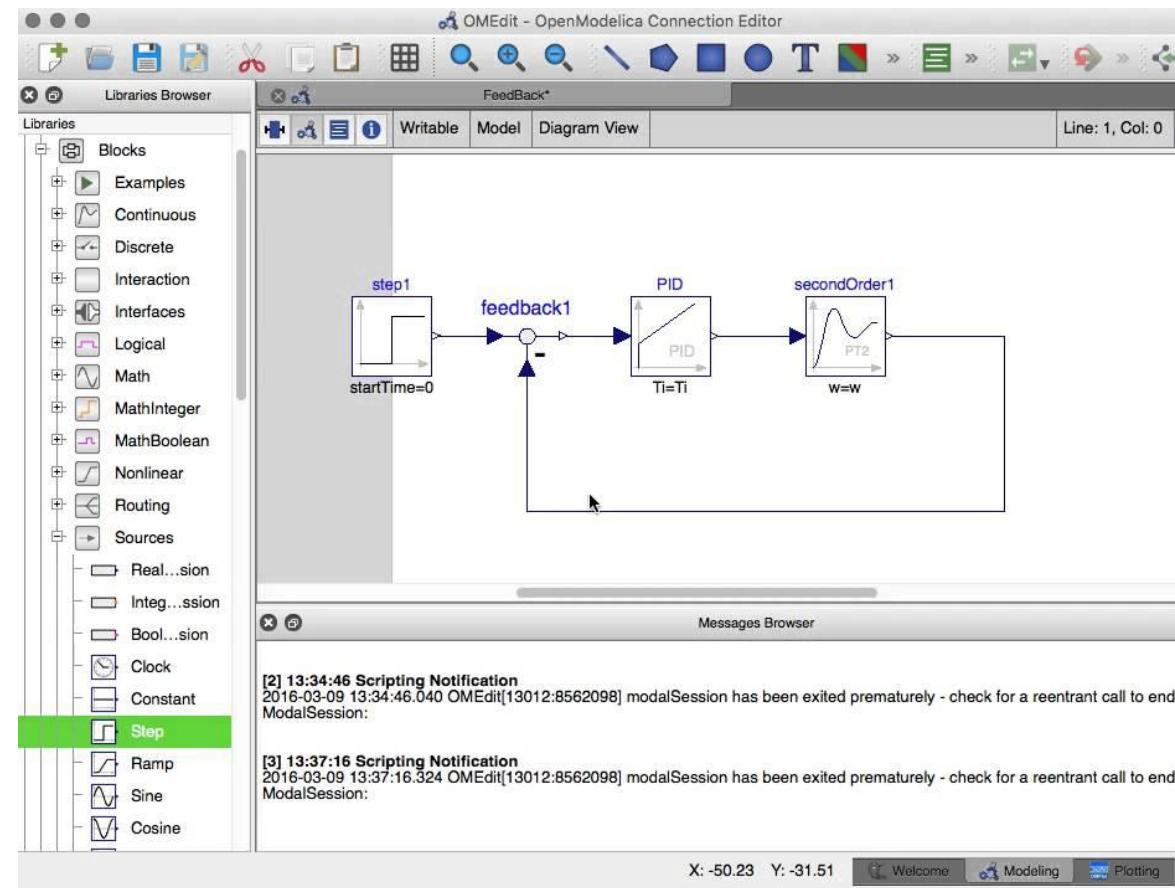
- Modelar ruidos de datos.
- Realizar la simulación de Monte Carlo.
- Determinar intervalos de confianza de los parámetros.



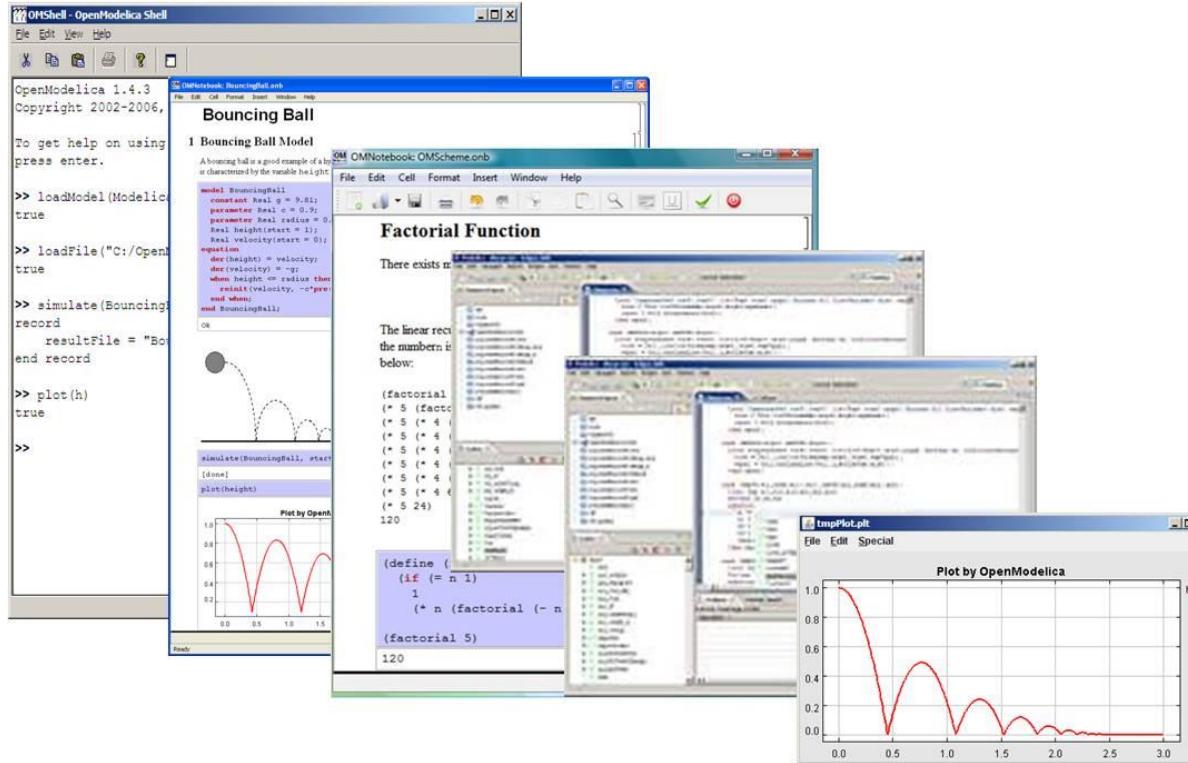
Open Modelica



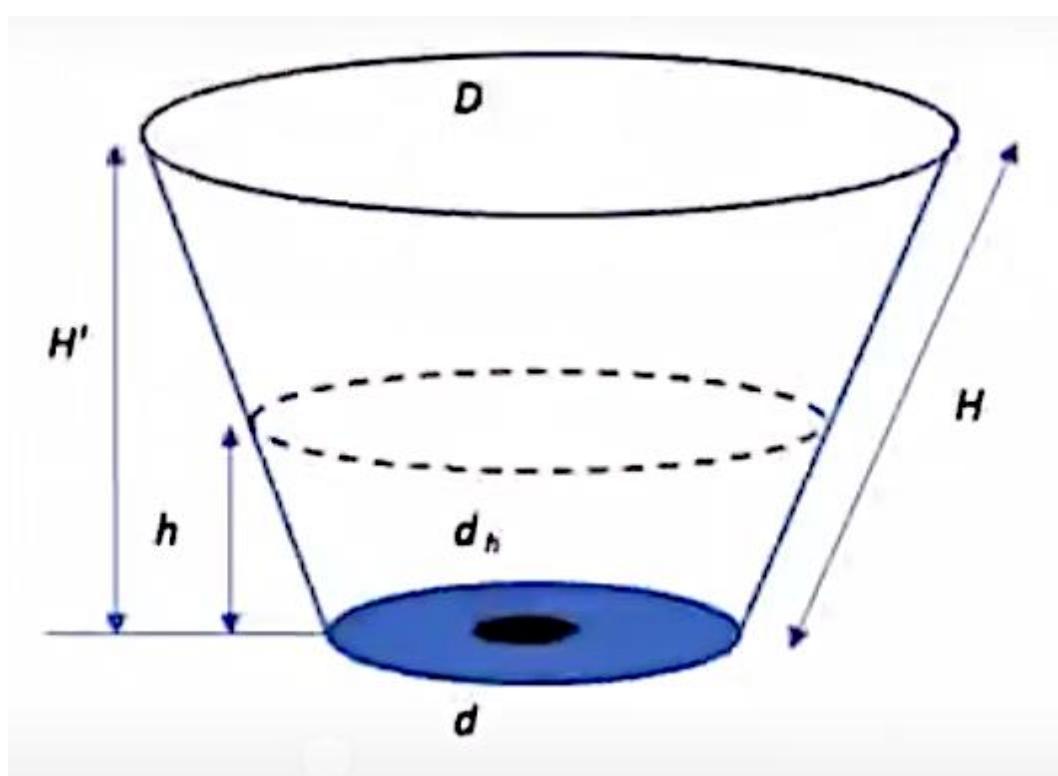
Open Modelica



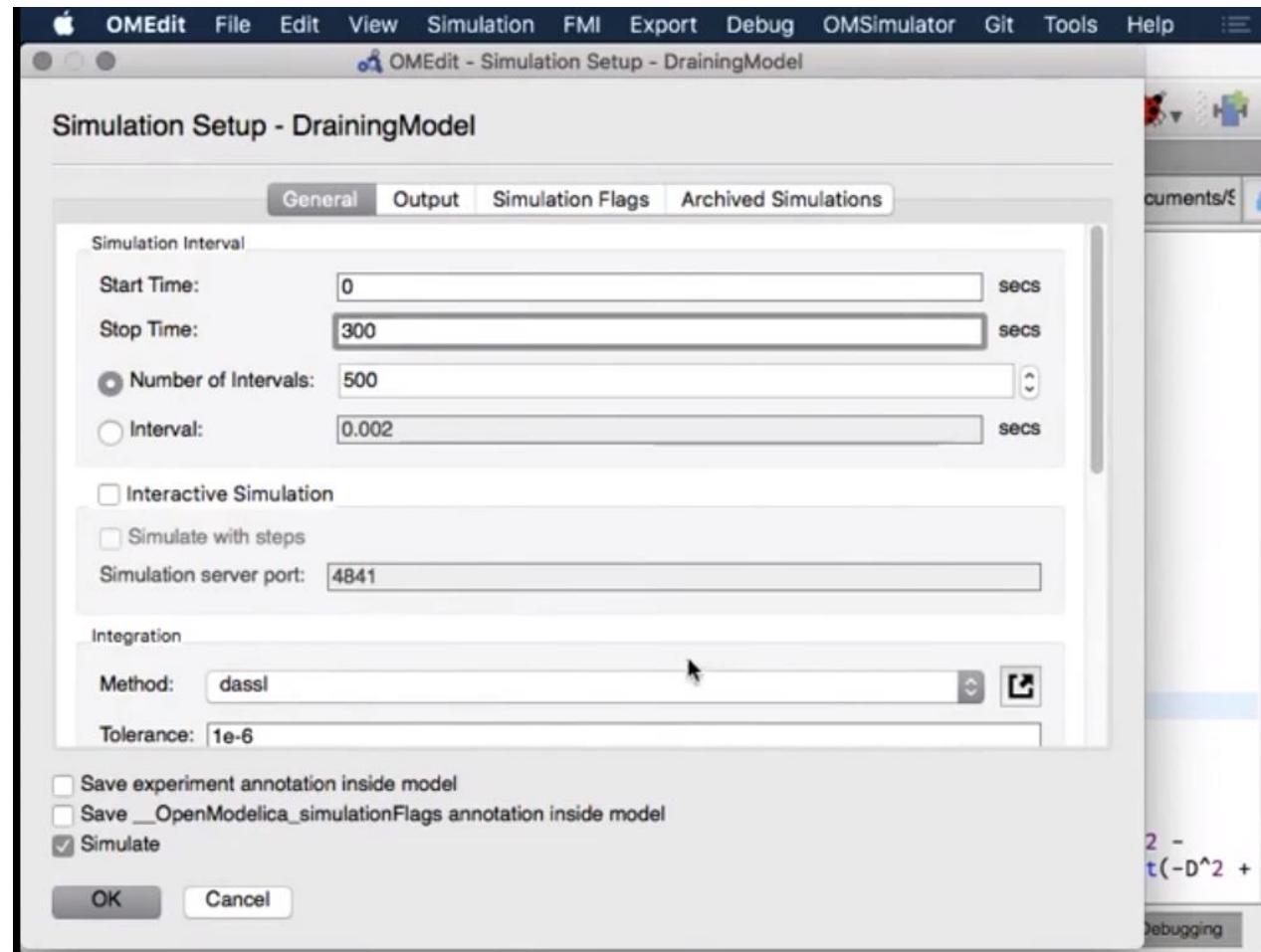
Open Modelica



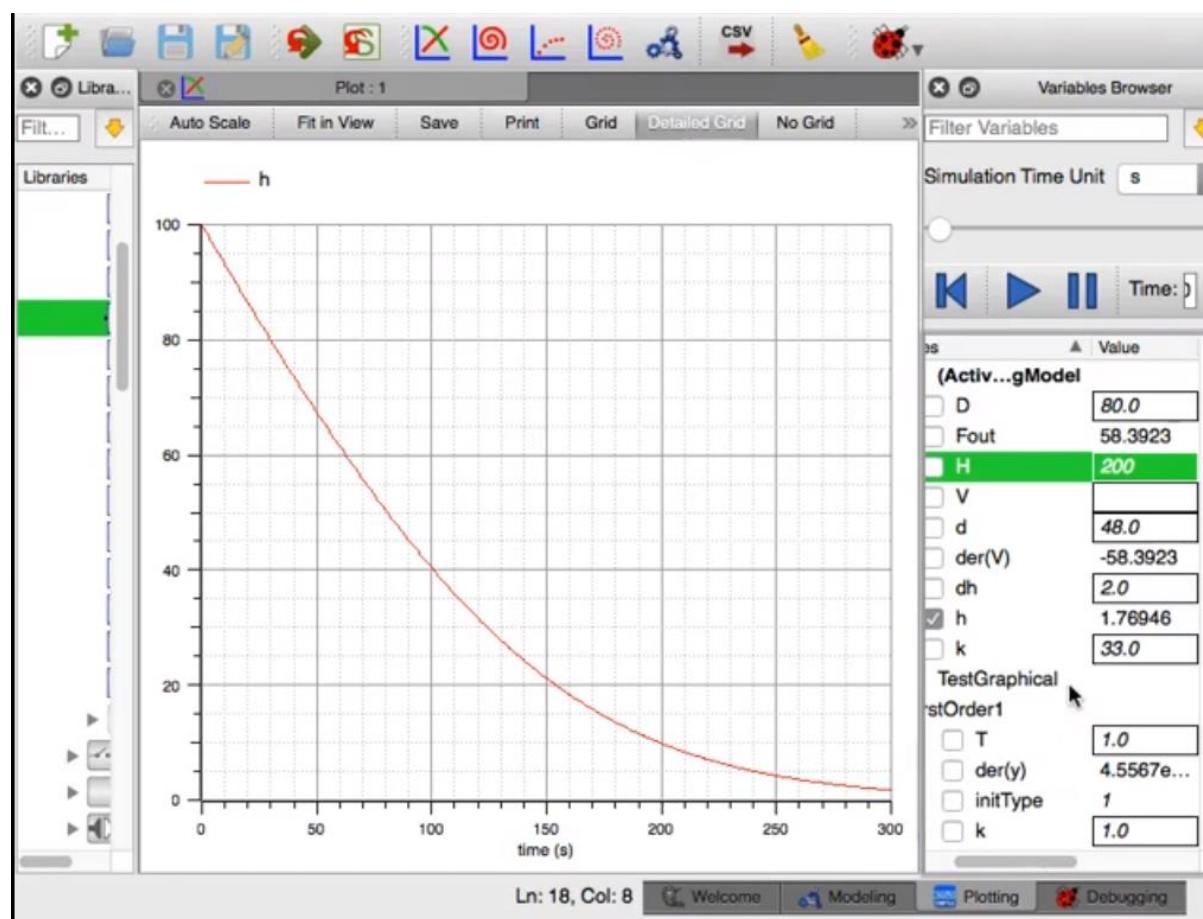
Tanque cónico



Open Modelica



Open Modelica



Sistema mezclador

Mixing system

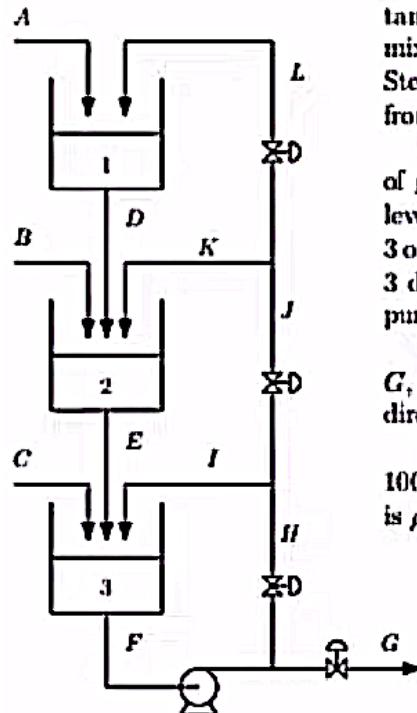


Figure 1 shows a set of well-mixed mixing tanks. All the streams contain a binary mixture of substance X and substance Y. Streams A, B and C are fed into the system from an upstream process.

Tanks 1 and 2 are drained by the force of gravity (assume flow is proportional to level), while the pump attached to the tank 3 output is sized such that the level in tank 3 does not affect the flowrate through the pump.

You may assume that the valves in lines G, H, J and L can manipulate those flows directly.

The density of substance X is $\rho_X = 1000 \text{ kg/m}^3$ and the density of substance Y is $\rho_Y = 800 \text{ kg/m}^3$.

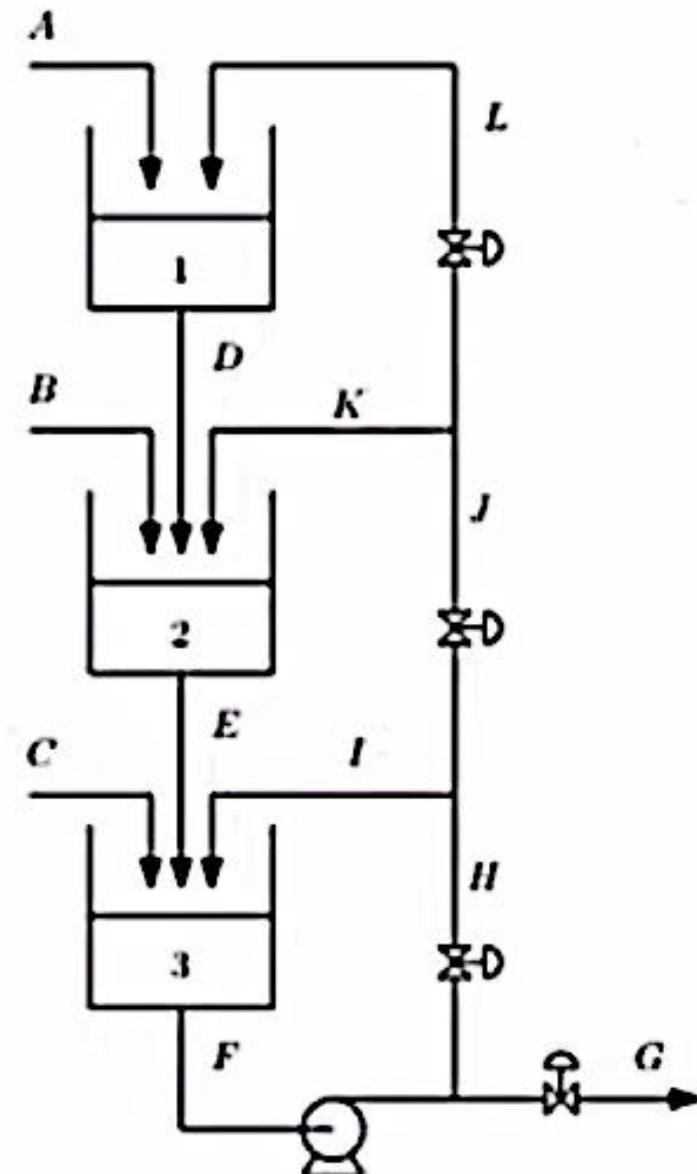
Figure 1: Mixing system

Sistema Mezclador

- Tanques bien agitados.
- Dos componentes: X e Y.
- Los caudales de descarga de los tanque 1 y 2 son proporcionales a los correspondientes niveles.
- Debido a la bomba de descarga, el caudal de descarga del tanque 3 es independiente del nivel.
- Las válvulas instaladas pueden manipular directamente los caudales G, H, J y L.
- La densidad de la substancia X es 1000 kg/m^3 y la densidad de la substancia Y es 800 kg/m^3 .

Sistema Mezclador

- Sistema isotérmico
- F es independiente de V_3 .
- Componentes: 2
- Balances dinámicos:
 - 1 global por tanque
 - 1 de componente por tanque
- Balances seudoestacionarios:
 - 1 global por divisor
 - 1 de componente por divisor



Análisis de grados de libertad

Users > alchemyst > Documents > Development > Dynamics-and-Control-admin > cpn321 > 2019_T3

```
1 model Tut3
2
3 // Flow rates, mass / time
4 Real A, B, C, D, E, F, G, H, I, J, K, L;
5 // Mass fraction X in each stream
6 Real xA, xB, xC, xD, xE, xF, xG, xH, xI, xJ, xK, xL;
7 // For each tank:
8 Real M1, M2, M3; // Mass
9 Real x1, x2, x3; // Mass fraction x
10 Real h1, h2, h3; // Height
11 Real V1, V2, V3; // Volume
12 parameter Real A1=3, A2=3, A3=3; // Cross-sectional area
13
14 parameter Real rhox=1000, rhoy=800; // Density of x and Y
15
16 // Setting fixed=false causes these values to be calculated at init
17 // Discharge coefficients
18 parameter Real k1(fixed=false), k2(fixed=false);
19 // Flowrates of the flows with valves
20 parameter Real Gstart(fixed=false), Hstart(fixed=false), Jstart(fixed=false);
21
22 initial equation
23
```



Users > alchemyst > Documents > Development > Dynamics-and-Control-admin > cpn321 > 2019_T3

```
17 // Discharge coefficients
18 parameter Real k1(fixed=false), k2(fixed=false);
19 // Flowrates of the flows with valves
20 parameter Real Gstart(fixed=false), Hstart(fixed=false), Jstart(fixed=false);
21
22 initial equation
23
24 h1 = 1;
25 h2 = 1;
26 h3 = 1;
27
28 H = G;
29 H = 2★J;
30 J = 2★L;
31
32 der(M1) = 0;
33 der(M2) = 0;
34 der(M3) = 0;
35
36 der(x1★M1) = 0;
37 der(x2★M2) = 0;
38 der(x3★M3) = 0;
```

40 equation

Users > alchemyst > Documents > Development > Dynamics-and-Control-admin > cpn321 > 2019_T3

```
40   equation
41
42   // Flowrates in
43   A = (if time ≤ 0 then 1 * rhox else 1.5 * rhox);
44   B = 1 * rhoy;
45   C = 1 * rhoz;
46
47   // Fractions in
48   xA = 1;
49   xB = 0;
50   xC = 0;
51   I
52   // Flowrates with valves
53   // Right now fixed at starting point, but could be changed here
54   G = Gstart; // For instance try this: G = A + B + C
55   H = Hstart;
56   J = Jstart;
57   L = Lstart;
58
59   // MB over tanks
60   der(M1) = A + L - D "MB Tank 1";
61   der(M2) = B + D + K - E "MB Tank 2";
62   der(M3) = C + E + I - F "MB Tank 3";
63
```



Users > alchemist > Documents > Development > Dynamics-and-Control-admin > cpn321 > 2019_T3

```
57 L = Lstart;
58
59 // MB over tanks
60 der(M1) = A + L - D "MB Tank 1";
61 der(M2) = B + D + K - E "MB Tank 2";
62 der(M3) = C + E + I - F "MB Tank 3";
63
64 // X balance over tanks
65 der(x1*M1) = xA*A + xL*L - xD*D "CB Tank 1";
66 der(x2*M2) = xB*B + xD*D + xK*K - xE*E "CB Tank 2";
67 der(x3*M3) = xC*C + xE*E + xI*I - xF*F "CB Tank 3";
68
69 //Junctions
70 F = H + G;
71 H = I + J;           I
72 J = K + L;
73
74 // Fractions are the same for all the junctions
75 xF = xG;
76 xF = xH;
77 xF = xI;
78 xF = xJ;
79 xF = xK;
80
```



Users > alchemist > Documents > Development > Dynamics-and-Control-admin > cpn321 > 2019_T3

```
80  xF = xL;
81
82  // Fractions coming out of tanks
83  xD = x1;
84  xE = x2;
85  xF = x3;
86
87  // Discharge
88  D = k1 * h1;
89  E = k2 * h2;
90
91  // Geometry
92  V1 = A1*h1;
93  V2 = A2*h2;
94  V3 = A3*h3;
95
96  // Mass to volume
97  V1 = M1*(x1/rhoX + (1 - x1)/rhoY);
98  V2 = M2*(x2/rhoX + (1 - x2)/rhoY);
99  V3 = M3*(x3/rhoX + (1 - x3)/rhoY);
100
101 annotation(
102   experiment(StartTime = 0, StopTime = 20, Tolerance = 1e-6, Inte
```

Aplicaciones imperativas

Sistema DAE

- Método robusto
- P : Vector de parámetros.
- U : Vector de variables manipulables.
- D : Vector de variables de perturbación.
- X : Vector de variables de estado.
- Y : Vector de variables de salida.
- t : Vector de tiempo.
- $tspan$: Intervalo de integración.
- X_0 : Vector de valores iniciales de X .

```
% Integración de las ODES
[t X] = ode15s(odefun,tspan,X0)

[P U D Y] = AEs_solution(t,X)
...

% Cálculo de las ODES
function dXdt = odefun(t,X)
% X y dXdt son vectores columna

% Cálculo de las AEs para el valor
% puntual de t
[P U D Y] = AEs_solution(t,X)
dXdt = ...;
end

% Resolución del sistema AEs
function [P U D Y] = AEs_solution(t,X)
...
end
```

Sistema DAE

- Método específico de MATLAB y GNU OCTAVE
- Y : Vector de variables del sistema.
- t : Vector de tiempo.
- $tspan$: Intervalo de integración.
- Y_0 : Vector de valores iniciales de Y .
- M : Matriz de masa.

```
% Resolución del DAEs
M = [1 0 0; 0 1 0; 0 0 0]; % 2 ODE, 1 AE
options = odeset('Mass',M);
[t,Y] = ode15s(daefun,tspan,Y0,options);
...
```

```
% Evaluación del DAEs
function out = daefun(t,X)
% X y out son vectores columna
dXdt = ODEs;
residuos = AES; // Igualadas a cero.
out = [dXdt, residuos];
end
```

Ejemplo de sistema DAE

Ecuaciones

$$\frac{dy_1}{dt} = -0.04y_1 + 10^4 y_2 y_3$$

$$\frac{dy_2}{dt} = 0.04y_1 - 10^4 y_2 y_3 - 3 \times 10^7 y_2^2$$

$$0 = y_1 + y_2 + y_3 - 1$$

VARIABLES

- Variables de estado: y_1, y_2
- Variable de salida: y_3
- Matriz de masa:

$$M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Implementación en GNU Octave

Método robusto

```
% ODEs
function dX = ODEs(t,X)
[y1 y2] = num2cell(X'){:,:};
Y = AEs(t,X);
[y3] = num2cell(Y){:,:};

dy1 = -0.04*y1 + 1e4*y2*y3;
dy2 = 0.04*y1 - 1e4*y2*y3 - 3e7*y2^2;

dX = [dy1 dy2]';
endfunction % ODEs
-----


% AEs
function Y = AEs(t,X)
% Datos
[y1 y2] = num2cell(X'){:,:};
y3 = 1 - y1 - y2;

Y = [y3];
endfunction % AEs

===== Programa =====

% Parámetros de simulación
graphsys = "gnuplot";

% Inicialización
X0 = [1; 0];

% Resolución
tpts = [0 4*logspace(-6,6)];
[tpts,X] = ode15s(@ODEs,tpts,X0);

% Cálculo de las variables dependientes
n = size(tpts,1);
YY = zeros(n,3);

for i =1:n
YY(i,:) = AEs(tpts(i),X(i,:));
endfor
```

Método específico

```
% Sistema DAE
function out = robertsdae(t,y)
out = [-0.04*y(1) + 1e4*y(2).*y(3)
        0.04*y(1) - 1e4*y(2).*y(3) - 3e7*y(2).^2
        y(1) + y(2) + y(3) - 1 ];
end

% Resolución
y0 = [1; 0; 0];
tspan = [0 4*logspace(-6,6)];
M = [1 0 0; 0 1 0; 0 0 0];
options = odeset('Mass',M,'RelTol',1e-4,'AbsTol',[1e-6
1e-10 1e-6]);
[t,y] = ode15s(@robertsdae,tspan,y0,options);
```

```
% ODEs
function dX = ODEs(t,X)
[y1 y2] = num2cell(X'){1,:};
Y = AEs(t,X);
[y3] = num2cell(Y){1,:};
dy1 = -0.04*y1 + 1e4*y2*y3;
dy2 = 0.04*y1 - 1e4*y2*y3 - 3e7*y2^2;
dX = [dy1 dy2]';
endfunction % ODEs
%-----
% AEs
function Y = AEs(t,X)
[y1 y2] = num2cell(X'){1,:};
y3 = 1 - y1 - y2;
Y = [y3];
endfunction % AEs

===== Programa =====
% Inicialización
X0 = [1; 0];

% Resolución
tpts = [0 4*logspace(-6,6)]';
[tpts,X] = ode15s(@ODEs,tpts,X0);

% Cálculo de las variables dependientes
n = size(tpts,1); YY = zeros(n,7);
for i =1:n
    YY(i,:) = AEs(tpts(i),X(i,:)');
endfor
```

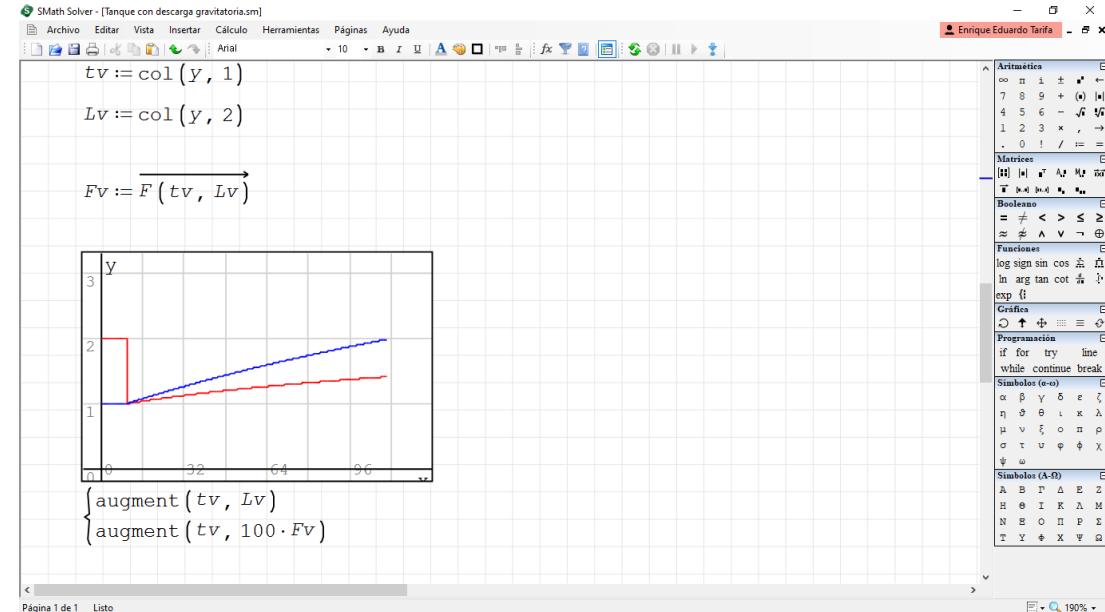
Implementación en GNU Octave

```
% Sistema DAE
function out = robertsdae(t,y)
out = [-0.04*y(1) + 1e4*y(2).*y(3)
        0.04*y(1) - 1e4*y(2).*y(3) - 3e7*y(2).^2
        y(1) + y(2) + y(3) - 1 ];
end

% Resolución
y0 = [1; 0; 0];
tspan = [0 4*logspace(-6,6)];
M = [1 0 0; 0 1 0; 0 0 0];
options = odeset('Mass',M,'RelTol',1e-4,'AbsTol',[1e-6 1e-10 1e-6]);
[t,y] = ode15s(@robertsdae,tspan,y0,options);
```

SMath Studio

- Notación matemática
- Unidades
- Vínculo con wxMaxima
- Controles
- Exportación de ejecutable
- Lento



Tanque con descarga gravitatoria

Parámetros del sistema

$$F_0 := 2 \cdot 10^{-3}$$

$$A := 0.785$$

$$C_V := 4.039 \cdot 10^{-5}$$

$$\rho := 1000$$

$$g := 9.81$$

Modelo del sistema

$$x(t) := \begin{cases} 0.5 & t < 100 \\ 0.25 & \text{else} \end{cases}$$

AEs

$$F(t, L) := C_V \cdot x(t) \cdot \sqrt{\rho \cdot g \cdot L}$$

ODEs

$$dL(t, L) := \frac{F_0 - F(t, L)}{A}$$

Simulación del tanque

$$D(t, X) := \begin{bmatrix} L \\ dL(t, L) \end{bmatrix} := X$$

$$X_0 := [1]$$

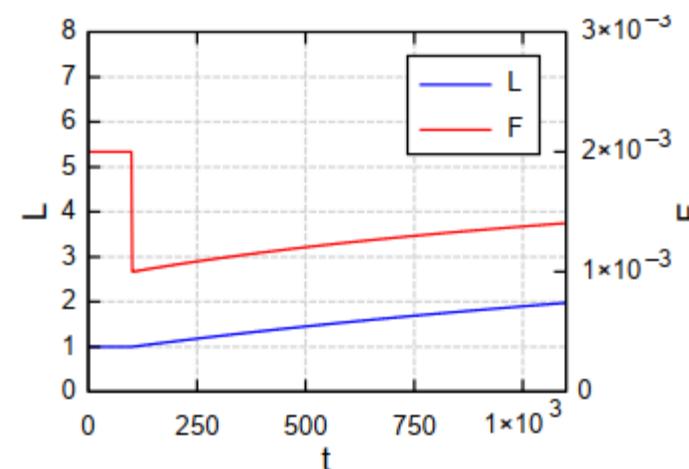
$$t_0 := 0 \quad t_f := 1100 \quad N := 500$$

$$y := \text{Rkadapt}(X_0, t_0, t_f, N, D)$$

$$tv := \text{col}(y, 1)$$

$$Lv := \text{col}(y, 2)$$

$$Fv := \overrightarrow{F(tv, Lv)}$$



$$\begin{cases} \text{augment}(tv, Lv) \\ \text{augment}(tv, Fv) \end{cases}$$

Euler Math Toolbox

- Interfaz con celdas
- Vínculo con wxMaxima
- Comentarios con LaTeX
- Sintaxis similar a MATLAB
- Unidades

You can size and position the graphics window as you like. Euler will remember the screen layout.

There is a special mode of Euler, where the plot appears in the text window. To see the plot, you then need to press the tabulator key.

Plotting Functions in one Variable

Euler can plot expressions, functions or data in 2D or 3D. 2D plots are generated with `plot2d`, and 3D plots with `plot3D`.

The easiest way to plot uses expressions. Expressions are strings containing an expression in the variable "x".

```
>plot2d("sin(x)/x",-2pi,2pi);
```

Press the tabulator key to see the plot, if the windows is hidden.

Of course, Euler has functions like `sinc(x)` which is defined as

$$\text{sinc}(x) = \begin{cases} \frac{\sin(x)}{x} & x \neq 0 \\ 0 & x = 0 \end{cases}$$

So this example could also have been plotted using "`sinc(x)`".

More functions can be added to the same plot using the parameter `>add`. The colors are set with "`color=...`". There are 16 predefined colors. But Euler can use any color with `rgb(red,green,blue)`.

Note the `:` after the following plot! This inserts the current graphics into the notebook after the command. These images will be saved in PNG format, and they will be used in the HTML export.

```
>plot2d("1-x^2/3!+x^4/5!",color=cyan,>add):
```

Euler Math Toolbox

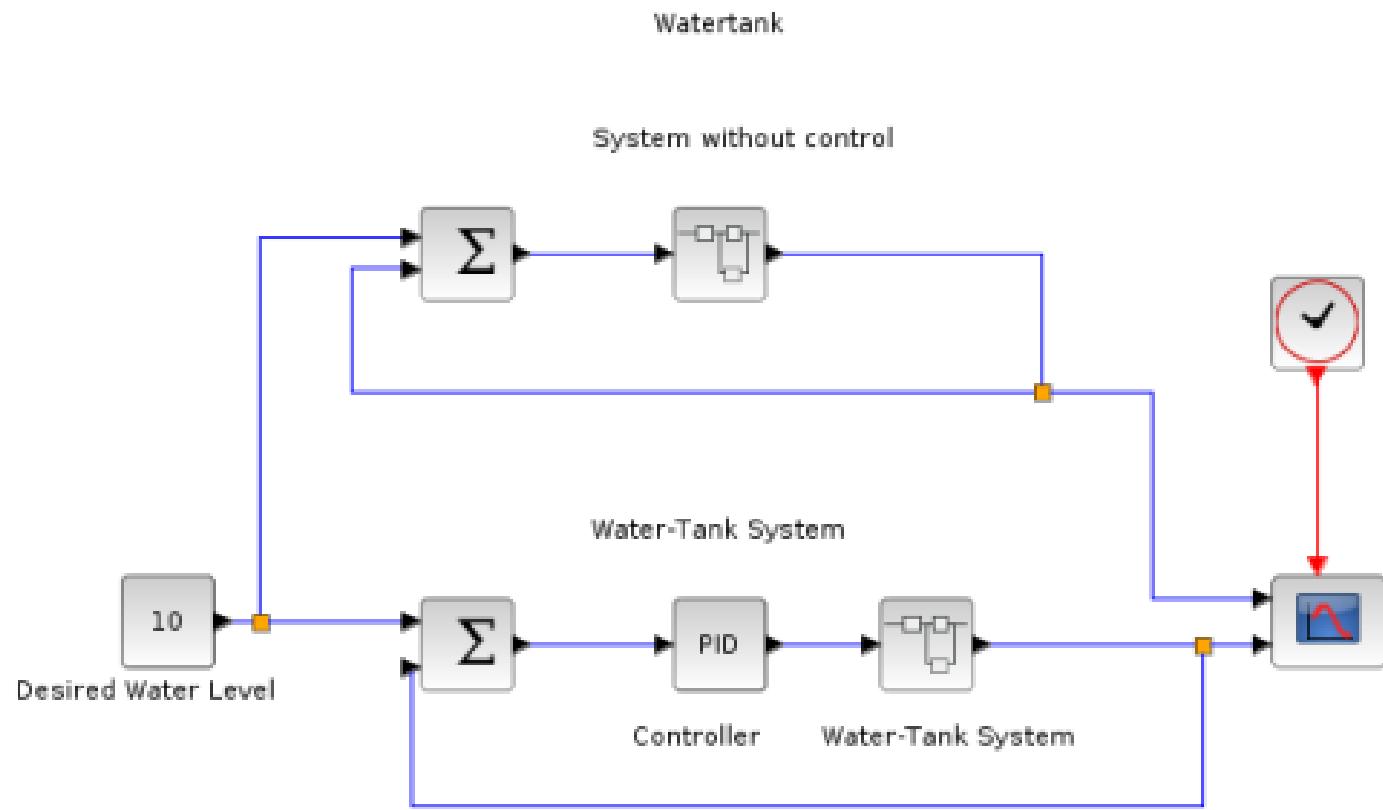
Tanque con descarga gravitatoria

```
>F0 = 20E-3; A = 0.785; Cv = 4.039E-4; rho = 1000; g = 9.81;
>function x(t) ...
    if t < 10 then return 0.5;
    else return 0.25;
  endif
endfunction

>function F(t,L) &= Cv*x(t)*sqrt(rho*g*L);
>function dL(t,L) &= (F0-F(t,L))/A;
>t = 0:1:110; L0 = 1;
>y = runge("dL",t,L0);
>plot2d(t,y,title="Nivel",xl="t",yl="L");
>plot2d(t,map(F,t,y),title="Descarga",xl="t",yl="F");
>
```

Scilab

- Cálculo numérico
- Orientado a matrices
- Sintaxis similar a MATLAB
- XCOS
- ODEs
- AEs



```

// Tanque con descarga gravitatoria
clear all;
close(winsid());
clc;

// ODEs
function dX = ODEs(t,X)
Y = AEs(t,X);
A = Y(1); Cv = Y(2); rho = Y(3); g = Y(4);
x = Y(5); F0 = Y(6); F = Y(7);

dL = (F0-F)/A;

dX = [dL];
endfunction //ODEs
//-----
//AEs
function Y = AEs(t,X)
// Datos
F0 = 20E-3; A = 0.785; Cv = 4.039E-4;
rho = 1000; g = 9.81; x = 0.5;

if t < 10
x = 0.5;
else
x = 0.25;
end

L = X(1);

F = Cv*x*sqrt(rho*g*L);

Y = [A Cv rho g x F0 F];
endfunction //AEs
//===== Programa =====
// Parámetros de simulación
tfin = 110; nts = 100;

// Inicialización
L0 = 1; X0 = [L0];
tpts = linspace(0, tfin, nts)';

// Resolución
X = ode(X0,0,tpts,ODEs)';

YY = zeros(nts,7);

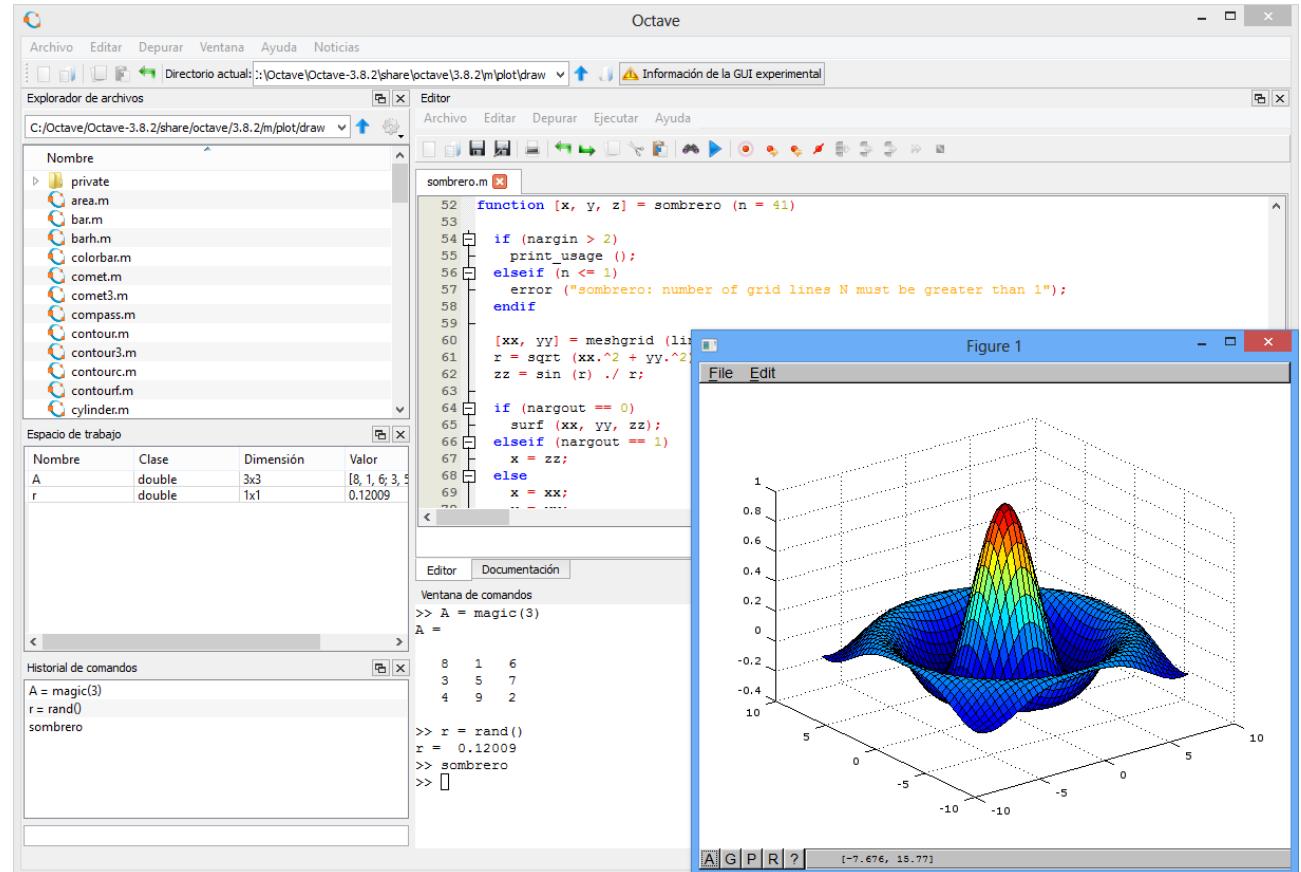
// Cálculo de las variables dependientes
for i = 1:nts
YY(i,:) = AEs(tpts(i),X(i));
end

// Gráfica
// Y = [A Cv rho g x F0 F];
figure(1);
plot(tpts,X(:,1)," -o");
figure(2);
plot(tpts,YY(:,7)," -o");
figure(3);
plot(tpts,YY(:,5)," -o");

```

GNU Octave

- Cálculo numérico
- Orientado a matrices
- Sintaxis similar a MATLAB
- Muy compatible con MATLAB
- ODEs
- AEs



```

% Tanque con descarga gravitatoria
clear all; close all; clc;

% ODEs
function dX = ODEs(t,X)
Y = AEs(t,X);
[A Cv rho g x F0 F] = num2cell(Y){1,:};

dL =(F0-F)/A;

dX = [dL];
endfunction % ODEs
%-----
% AEs
function Y = AEs(t,X)
% Parámetros
F0 = 20E-3; A = 0.785; Cv = 4.039E-4;
rho = 1000; g = 9.81;

if t < 10
x = 0.5;
else
x = 0.25;
endif

L = X(1);

F = Cv*x*sqrt(rho*g*L);

Y = [A Cv rho g x F0 F];
endfunction % AEs

```

```

%===== Programa =====
% Parámetros de simulación
graphsys = "gnuplot";
tfin = 110; nts = 100;
adaptativo = false; % paso constante o variable

% Inicialización
L0 = 1;
X0 = [L0];

% Resolución
if ~adaptativo
tpts = linspace(0, tfin, nts)';
[tpts,X] = ode15s(@ODEs,tpts,X0);
else % adaptativo
[tpts,X] = ode15s(@ODEs,[0 tfin],X0);
endif

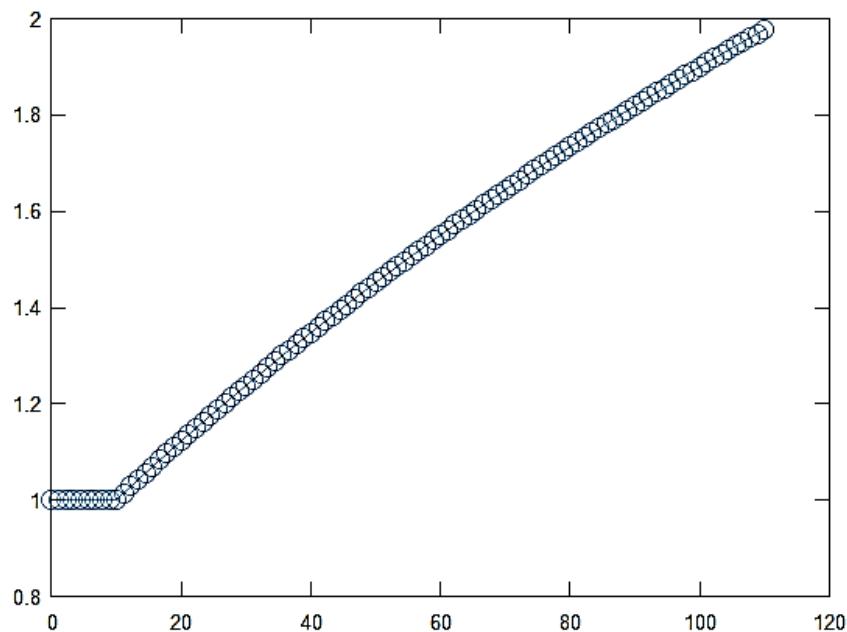
% Cálculo de las variables dependientes
n = size(tpts,1);
Y = zeros(n,7);
for i = 1:n
Y(i,:) = AEs(tpts(i),X(i));
endfor

% Gráfica
% Y = [A Cv rho g x F0 F];
graphics_toolkit(graphsys);
figure(1);
plot(tpts,X(:,1),"o");

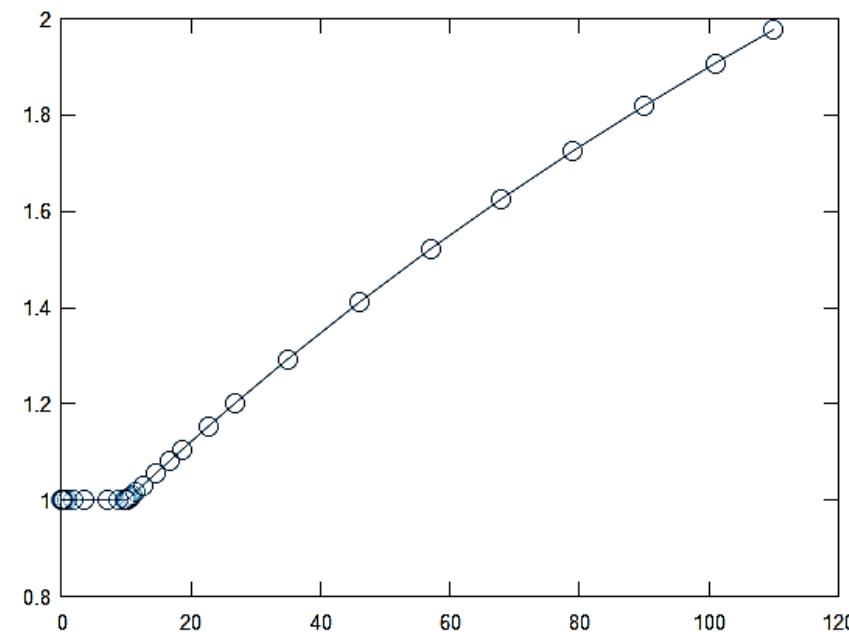
```

Método adaptativo para el nivel

Sin adaptación



Con adaptación



Julia

- Lenguaje de propósito general
- Maneja matrices
- Sintaxis similar a MATLAB
- Compilación especial JIT
- ODEs
- AEs

The screenshot shows a Visual Studio Code interface with a Julia script named `tanque.jl` open in the editor. The code defines a function `F(L)` for Adiabatic Evaporation (AE) and a function `tanque` for solving an ordinary differential equation (ODE). It uses the `ODEProblem` and `solve` functions from the DifferentialEquations package. The script also includes code to plot the results.

```
File Edit Selection View Go Run Terminal Help
tanque.jl <...>
tanque.jl > ...
24
25 #AEs
26 F(L) = Cv*x*sqrt(rho*g*L)
27
28 #ODEs
29 tanque = @ode_def begin
30 | dL = (F0-F(L))/A
31 end F0 A
32
33 prob = ODEProblem(tanque,x0,tspan,p)
34 sol = solve(prob,Tsit5())
35
36 #Evito que se sobreescriba el gráfico
37 p = plot(sol,vars=(1))
38 #plot!(p,sol,vars=(2))
39 display(p)
40
```

PROBLEMS 27 OUTPUT DEBUG CONSOLE TERMINAL

```
t: 6-element Vector{Float64}:
 0.0
  :
 110.0
u: 6-element Vector{Vector{Float64}}:
 [1.0]
```

Julia Plots (4/4) - Julia - Visual Studio Code

The right side of the interface shows a plot titled "Julia Plots (4/4)" with a single data series labeled "F". The plot shows a curve starting at approximately (0, 0.010) and increasing monotonically to about (100, 0.014).

```

#Listado en Julia
#Tanque con descarga gravitatoria

using ParameterizedFunctions, Plots,
OrdinaryDiffEq

#Condiciones iniciales
L0 = 1
x0 = [L0]

#Rango de tiempo
tspan = (0.0,110.0)

#Parámetros
F0 = 20E-3
A = 0.785
p = [F0 A]
Cv = 4.039E-4
rho = 1000
g = 9.81
x = 0.25

```

```

#AEs
F(L) = Cv*x*sqrt(rho*g*L)

#ODEs
tanque = @ode_def begin
    dL = (F0-F(L))/A
end F0 A

prob = ODEProblem(tanque,x0,tspan,p)
sol = solve(prob,Tsit5())

#Evito que se sobreescriba el gráfico
p = plot(sol,vars=(1))
#plot!(p,sol,vars=(2))
display(p)

#Para suavizar el plot de F
#Si no se hace esto, grafica 6 puntos.
t = range(0,sol.t[end],20)
plot(t,map(F,sol(t)[1,:]),label="F")

```