Desarrollo de una versión de UNAMALLA en Scilab

Seminario Laboratorio Computo Científico
FC UNAM

César Carreón Otañez

30 de septiembre de 2009

“Software Libre”

- Años 60,70, acceso-intercambio a códigos-programas.
- En los años 80’s comienzan a aparecer computadoras con Sistemas Operativos privados (códigos cerrados, imposición).
  - Se impide realizar modificaciones al programa.
  - En caso de hallar errores sólo se reportaban.

“Software Libre”

- A principios de los años 80’s nace el concepto de *Free Software*. MIT *Massachusetts Institute of Technology*.
- En 1984, Richard Stallman comenzó a trabajar en el proyecto GNU y un año más tarde fundó la Free Software Foundation (FSF).
- Stallman introdujo la definición de free software y el concepto de “copyleft”.

El software libre es una cuestión de libertad, no de precio

Cuatro libertades.

1. La libertad de ejecutar el programa, para cualquier propósito.

El software libre es una cuestión de libertad, no de precio

Cuatro libertades.

2. La libertad de estudiar cómo trabaja el programa, y cambiarlo para que haga lo que usted quiera. Acceso al código fuente.

El software libre es una cuestión de libertad, no de precio

Cuatro libertades.

3. La libertad de redistribuir copias para que pueda ayudar al prójimo.

El software libre es una cuestión de libertad, no de precio

Cuatro libertades.

4. La libertad de mejorar el programa y publicar sus mejoras, y versiones modificadas en general, para que se beneficie toda la comunidad. Acceso al código fuente.

Ejemplos

1. Sistemas operativos: Linux (Debian, Red Hat, etc.).

Ejemplos

1. Sistemas operativos: Linux (Debian, Red Hat, etc.).
2. Interfaces gráficas de usuarios: GNome, KDE.

Ejemplos

1. Sistemas operativos: Linux (Debian, Red Hat, etc.).
2. Interfaces gráficas de usuarios: GNome, KDE.
3. Navegadores: Mozilla, Firefox.

Ejemplos

1. Sistemas operativos: Linux (Debian, Red Hat, etc.).
2. Interfaces gráficas de usuarios: GNome, KDE.
3. Navegadores: Mozilla, Firefox.
4. OpenOffice, Maxima, Scilab.

Scilab

- Lenguaje para cálculo científico.

Scilab

- Unix, GNU/Linux, Windows, Solaris, Alpha.

Scilab

- Desarrollado por INRIA (Institut National de Recherche en Informatique et Automatique) y la ENPC (École Nationale des Ponts et Chaussées) desde 1990.

Scilab

- Interacción con lenguajes (FORTRAN, Java y C y C++).

Herramientas

- Gráficos 2-D, 3-D, animación. Álgebra lineal.

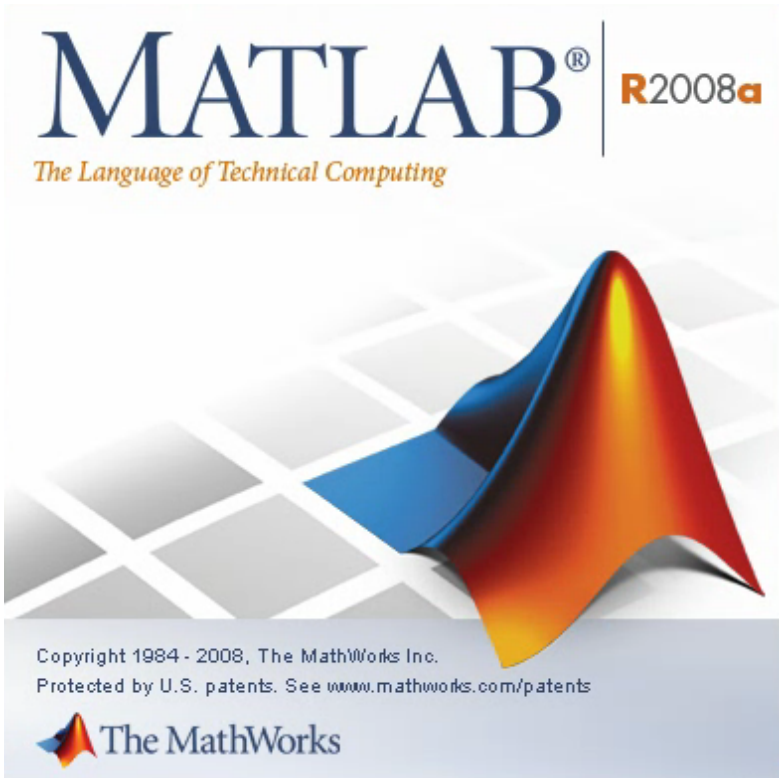
Herramientas

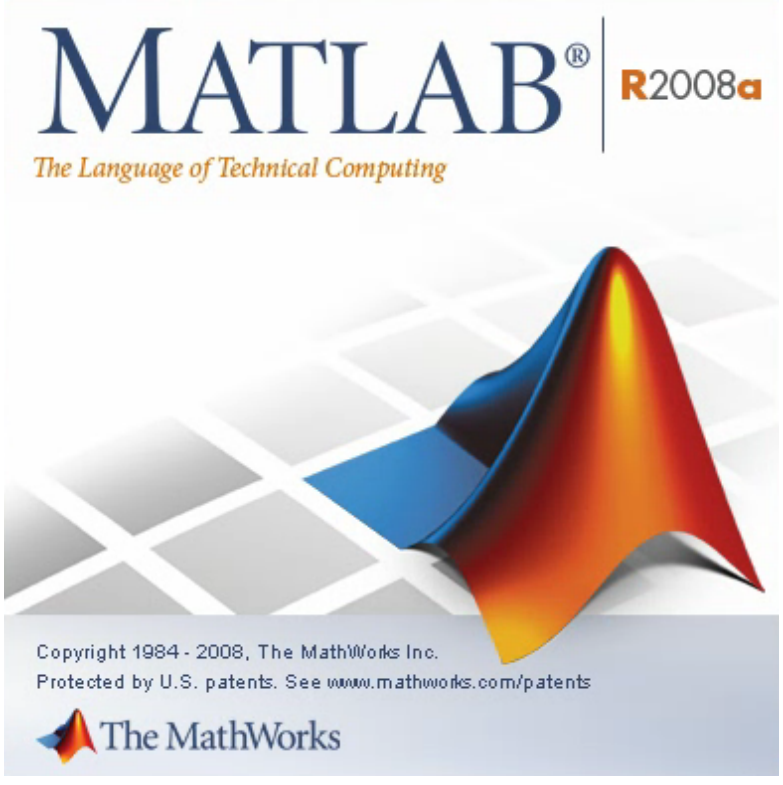
- Simulación Sistemas Ecuaciones Diferenciales (Explícitas/ Implícitas).

Herramientas

- Grafos y Redes, Estadística, Interfaz con cálculo simbólico.

Scilab - Matlab

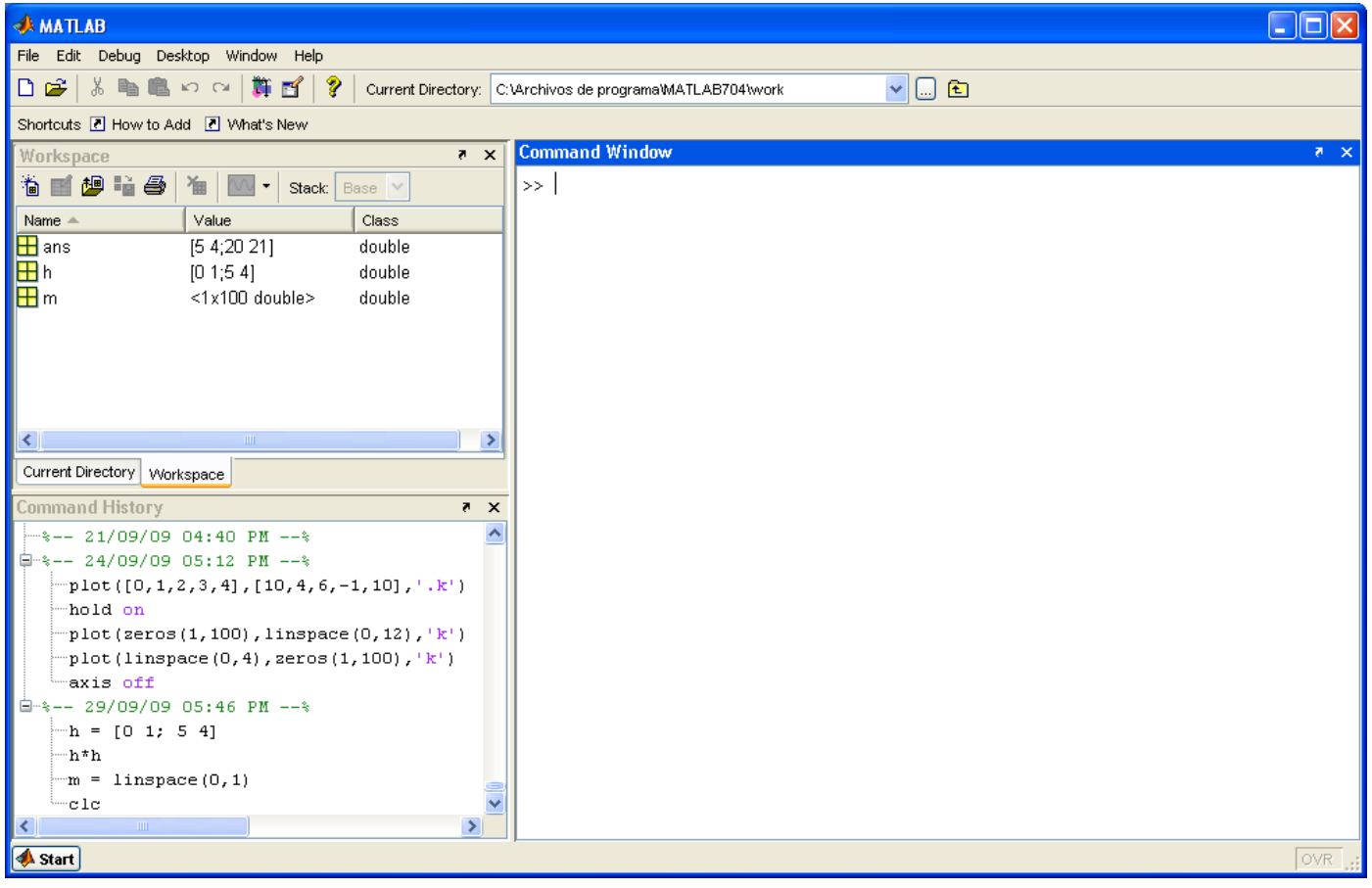




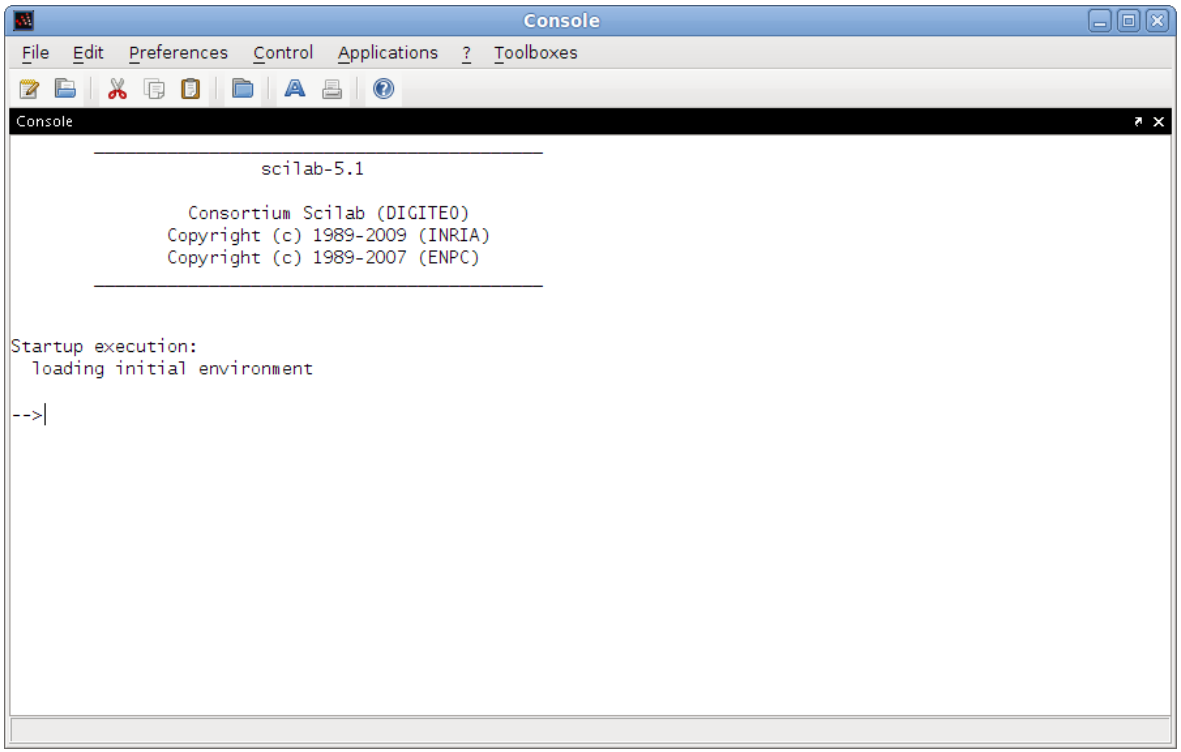
- Pago por la licencia.
- Código cerrado.

- Software libre.
- Código.
- Tiempo de ejecución.

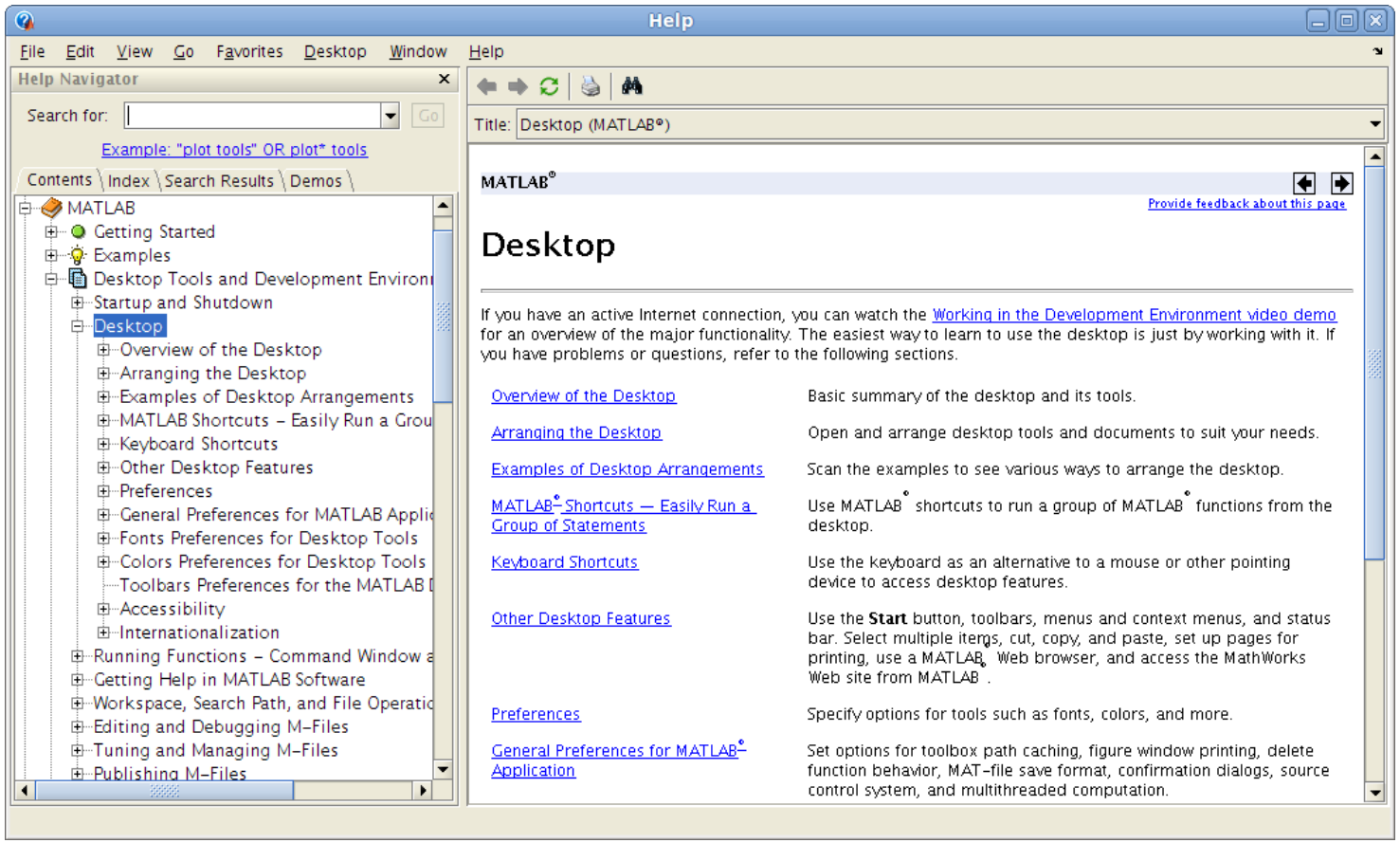
GUI


MATLAB
File Edit Debug Desktop Window Help
Current Directory: C:\Archivos de programa\MATLAB704\work
Shortcuts How to Add What's New
Workspace
Stack: Base
Name Value Class
ans [5 4;20 21] double
h [0 1;5 4] double
m <1x100 double> double
Current Directory Workspace
Command History
%-- 21/09/09 04:40 PM --%
%-- 24/09/09 05:12 PM --%
plot([0,1,2,3,4],[10,4,6,-1,10],'.k')
hold on
plot(zeros(1,100),linspace(0,12),'k')
plot(linspace(0,4),zeros(1,100),'k')
axis off
%-- 29/09/09 05:46 PM --%
h = [0 1; 5 4]
h*h
m = linspace(0,1)
clc
Command Window
>>
Start
OVR

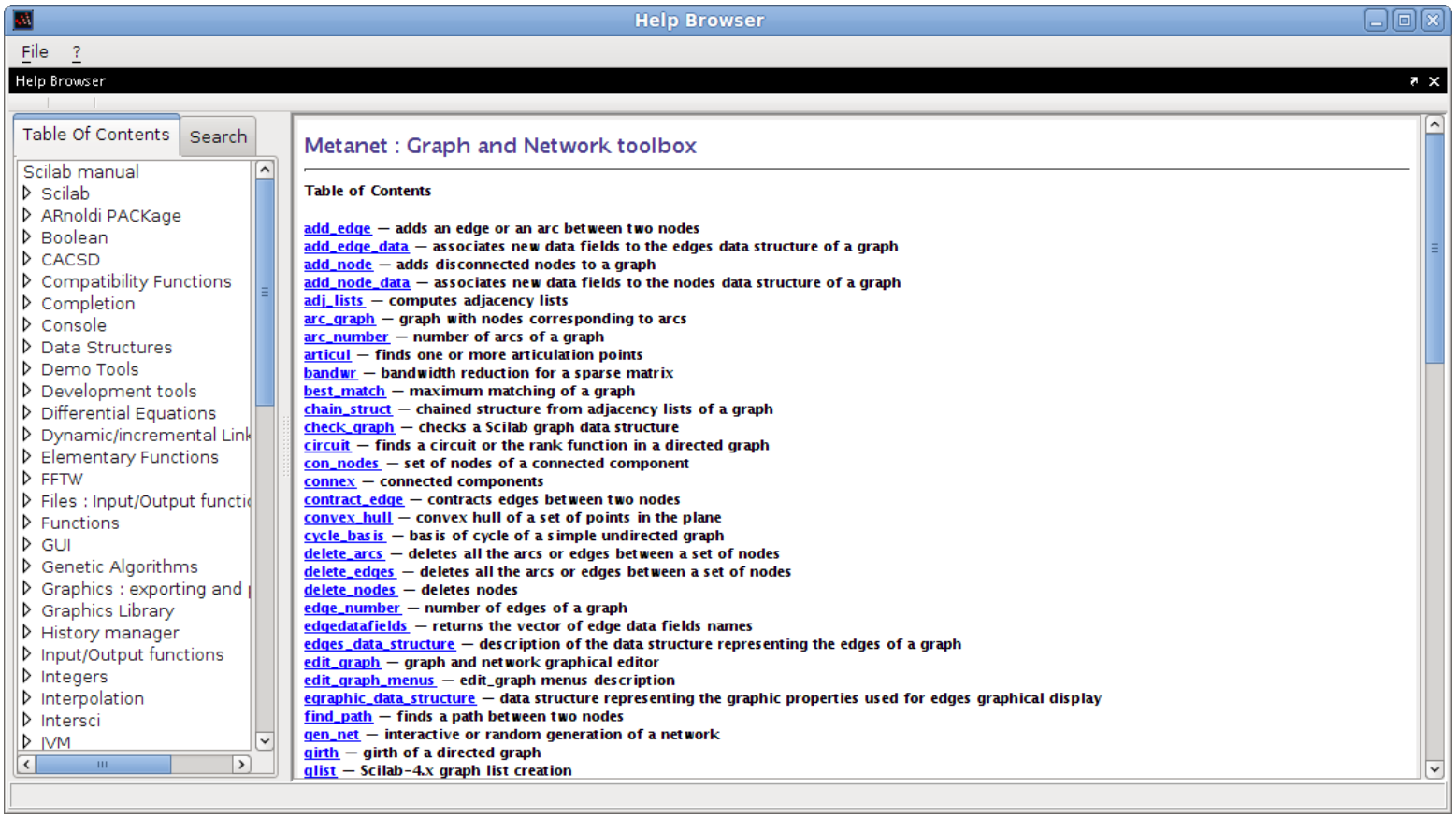
GUI


Console
File Edit Preferences Control Applications ? Toolboxes
Console
scilab-5.1
Consortium Scilab (DIGITEO)
Copyright (c) 1989-2009 (INRIA)
Copyright (c) 1989-2007 (ENPC)
Startup execution:
loading initial environment
-->

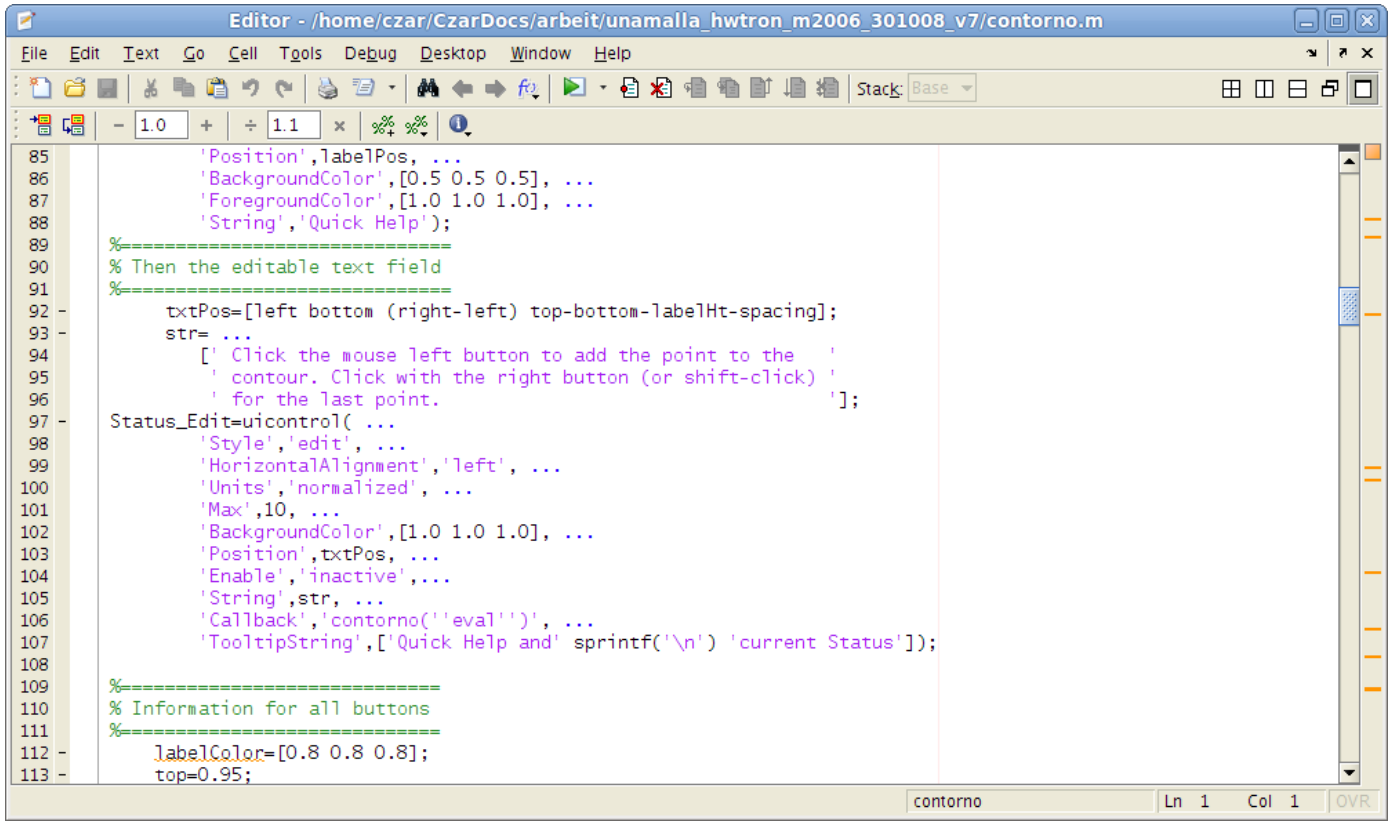
Help


Help
File Edit View Go Favorites Desktop Window Help
Help Navigator
Search for:
Go
Example: "plot tools" OR plot* tools
Contents Index Search Results Demos
MATLAB
Getting Started
Examples
Desktop Tools and Development Environ
Startup and Shutdown
Desktop
Overview of the Desktop
Arranging the Desktop
Examples of Desktop Arrangements
MATLAB Shortcuts – Easily Run a Grou
Keyboard Shortcuts
Other Desktop Features
Preferences
General Preferences for MATLAB Appli
Fonts Preferences for Desktop Tools
Colors Preferences for Desktop Tools
Toolbars Preferences for the MATLAB
Accessibility
Internationalization
Running Functions – Command Window a
Getting Help in MATLAB Software
Workspace, Search Path, and File Operatio
Editing and Debugging M-Files
Tuning and Managing M-Files
Publishing M-Files
Title: Desktop (MATLAB®)
MATLAB®
Provide feedback about this page
Desktop
If you have an active Internet connection, you can watch the Working in the Development Environment video demo for an overview of the major functionality. The easiest way to learn to use the desktop is just by working with it. If you have problems or questions, refer to the following sections.
Overview of the Desktop
Basic summary of the desktop and its tools.
Arranging the Desktop
Open and arrange desktop tools and documents to suit your needs.
Examples of Desktop Arrangements
Scan the examples to see various ways to arrange the desktop.
MATLAB® Shortcuts — Easily Run a Group of Statements
Use MATLAB® shortcuts to run a group of MATLAB® functions from the desktop.
Keyboard Shortcuts
Use the keyboard as an alternative to a mouse or other pointing device to access desktop features.
Other Desktop Features
Use the Start button, toolbars, menus and context menus, and status bar. Select multiple items, cut, copy, and paste, set up pages for printing, use a MATLAB® Web browser, and access the MathWorks Web site from MATLAB®.
Preferences
Specify options for tools such as fonts, colors, and more.
General Preferences for MATLAB® Application
Set options for toolbox path caching, figure window printing, delete function behavior, MAT-file save format, confirmation dialogs, source control system, and multithreaded computation.

Help


Help Browser
File ?
Help Browser
Table Of Contents
Search
Scilab manual
Scilab
ARnoldi PACKage
Boolean
CACSD
Compatibility Functions
Completion
Console
Data Structures
Demo Tools
Development tools
Differential Equations
Dynamic/incremental Link
Elementary Functions
FFTW
Files : Input/Output functi
Functions
GUI
Genetic Algorithms
Graphics : exporting and
Graphics Library
History manager
Input/Output functions
Integers
Interpolation
Intersci
JVM
Metanet : Graph and Network toolbox
Table of Contents
add_edge — adds an edge or an arc between two nodes
add_edge_data — associates new data fields to the edges data structure of a graph
add_node — adds disconnected nodes to a graph
add_node_data — associates new data fields to the nodes data structure of a graph
adj_lists — computes adjacency lists
arc_graph — graph with nodes corresponding to arcs
arc_number — number of arcs of a graph
articul — finds one or more articulation points
bandwr — bandwidth reduction for a sparse matrix
best_match — maximum matching of a graph
chain_struct — chained structure from adjacency lists of a graph
check_graph — checks a Scilab graph data structure
circuit — finds a circuit or the rank function in a directed graph
con_nodes — set of nodes of a connected component
connex — connected components
contract_edge — contracts edges between two nodes
convex_hull — convex hull of a set of points in the plane
cycle_basis — basis of cycle of a simple undirected graph
delete_arcs — deletes all the arcs or edges between a set of nodes
delete_edges — deletes all the arcs or edges between a set of nodes
delete_nodes — deletes nodes
edge_number — number of edges of a graph
edgedatafields — returns the vector of edge data fields names
edges_data_structure — description of the data structure representing the edges of a graph
edit_graph — graph and network graphical editor
edit_graph_menus — edit_graph menus description
egraphic_data_structure — data structure representing the graphic properties used for edges graphical display
find_path — finds a path between two nodes
gen_net — interactive or random generation of a network
girth — girth of a directed graph
glist — Scilab-4.x graph list creation

Editor




```
85          'Position',labelPos, ...
86          'BackgroundColor',[0.5 0.5 0.5], ...
87          'ForegroundColor',[1.0 1.0 1.0], ...
88          'String','Quick Help');
89      %==============================
90      % Then the editable text field
91      %==============================
92 -        txtPos=[left bottom (right-left) top-bottom-labelHt-spacing];
93 -        str= ...
94             [' Click the mouse left button to add the point to the   '
95              ' contour. Click with the right button (or shift-click) '
96              ' for the last point.                                    '];
97 -    Status_Edit=uicontrol( ...
98          'Style','edit', ...
99          'HorizontalAlignment','left', ...
100         'Units','normalized', ...
101         'Max',10, ...
102         'BackgroundColor',[1.0 1.0 1.0], ...
103         'Position',txtPos, ...
104         'Enable','inactive',...
105         'String',str, ...
106         'Callback','contorno(''eval'')', ...
107         'TooltipString',['Quick Help and' sprintf('\n') 'current Status']);
108
109     %==============================
110     % Information for all buttons
111     %==============================
112 -       labelColor=[0.8 0.8 0.8];
113 -       top=0.95;
```

Editor

SciPad 7.18 - unamalla.sci

File Edit Search Execute Debug Scheme Options Windows Help

```
115  //
116
117       Menu_contour = uimenu( ...
118                              'parent'     , Main_panel, ...
119                              'label'      , gettext('Contour') ...
120                              );
121                      uimenu( ...
122                              'parent'     , Menu_contour, ...
123                               'label'      , gettext('Open'), ...
124                              'callback'   , 'unamalla(''Open_Contour'')' ...
125                              );
126                      // Menu New Polygon
127    NewPoly_contour = uimenu( ...
128                              'parent'     , Menu_contour, ...
129                              'label'      , gettext('New Polygon'), ...
130                              'enable'     , 'on', ...
131                              'callback'   , 'unamalla(''New_Polygon'')', ...
132                              'tag'        , 'New_Polygon' ...
133                               );
134                      // Menu Edit Boundaries
135      Edit_contour = uimenu( ...
136                              'parent'     , Menu_contour, ...
137                              'label'      , gettext('Edit Boundaries'), ...
138                              'enable'     , 'off', ...
139                              'callback'   , 'unamalla(''Edit'')', ...
140                              'tag'        , 'Edit' ...
141                               );
142                      // Menu Tratamiento Contour
143     Traetment_contour = uimenu( ...
144                              'parent'     , Menu_contour, ...
145                              'label'      , gettext('Traetment') ...
146                               );
147                                       // --------   REDUCTION  PROCESS ----------
148     Menu_Reduction = uimenu( ...
149                              'parent'     , Traetment_contour, ...
150                              'label'      , gettext('Reduction'), ...
```

Line: 1 Column: 1 | Logical line: 1

UNAMALLA en Scilab

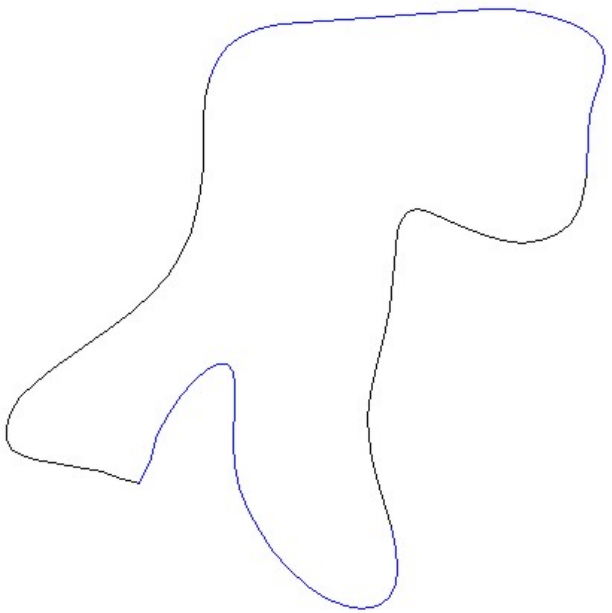
¿Para qué?

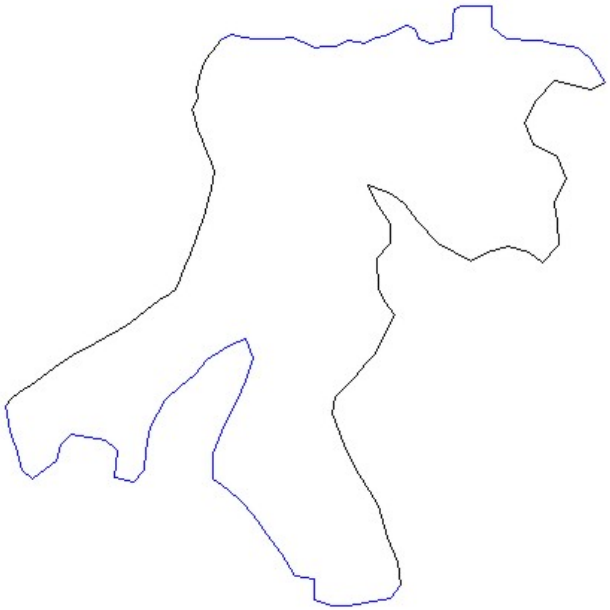
- Es una técnica utilizada como parte del tratamiento de Contornos (UNAMALLA).
  1. Reducción de Puntos-“Repoblación”.
  2. SUAVIZAMIENTO CÓNICO.
  3. Reparametrización.
- En ocasiones ayuda a mejorar las propiedades de la malla.

Premisa Central

CONSERVAR LO MÁS POSIBLE LA FORMA DEL CONTORNO ORIGINAL.

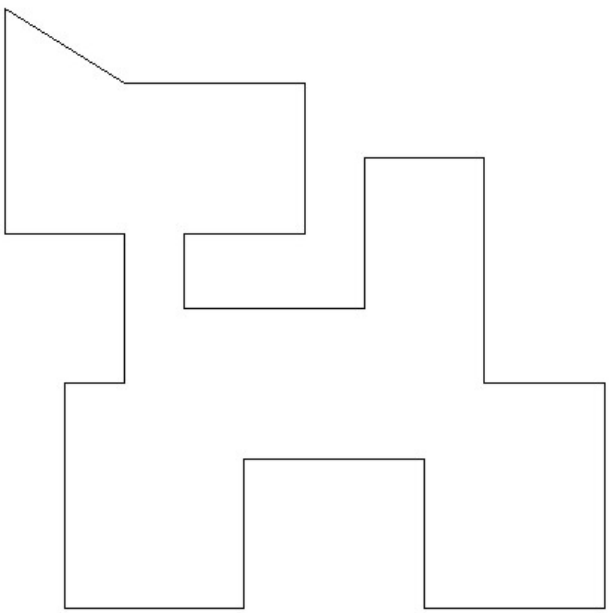
Spline Cónico vs Cúbico

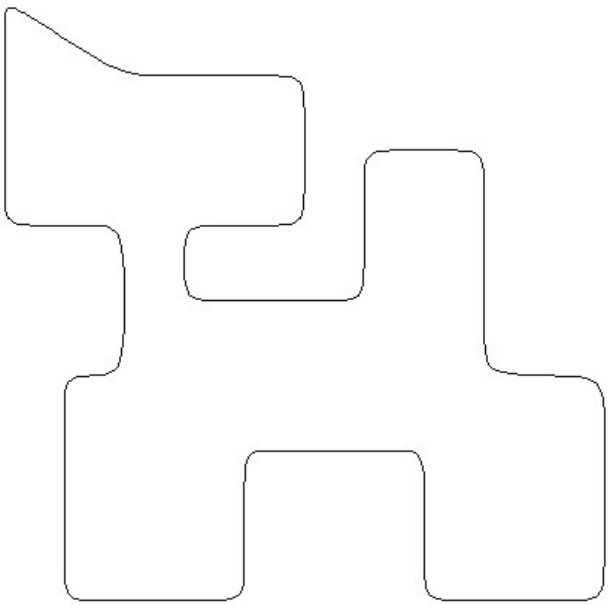




Idea Geométrica.

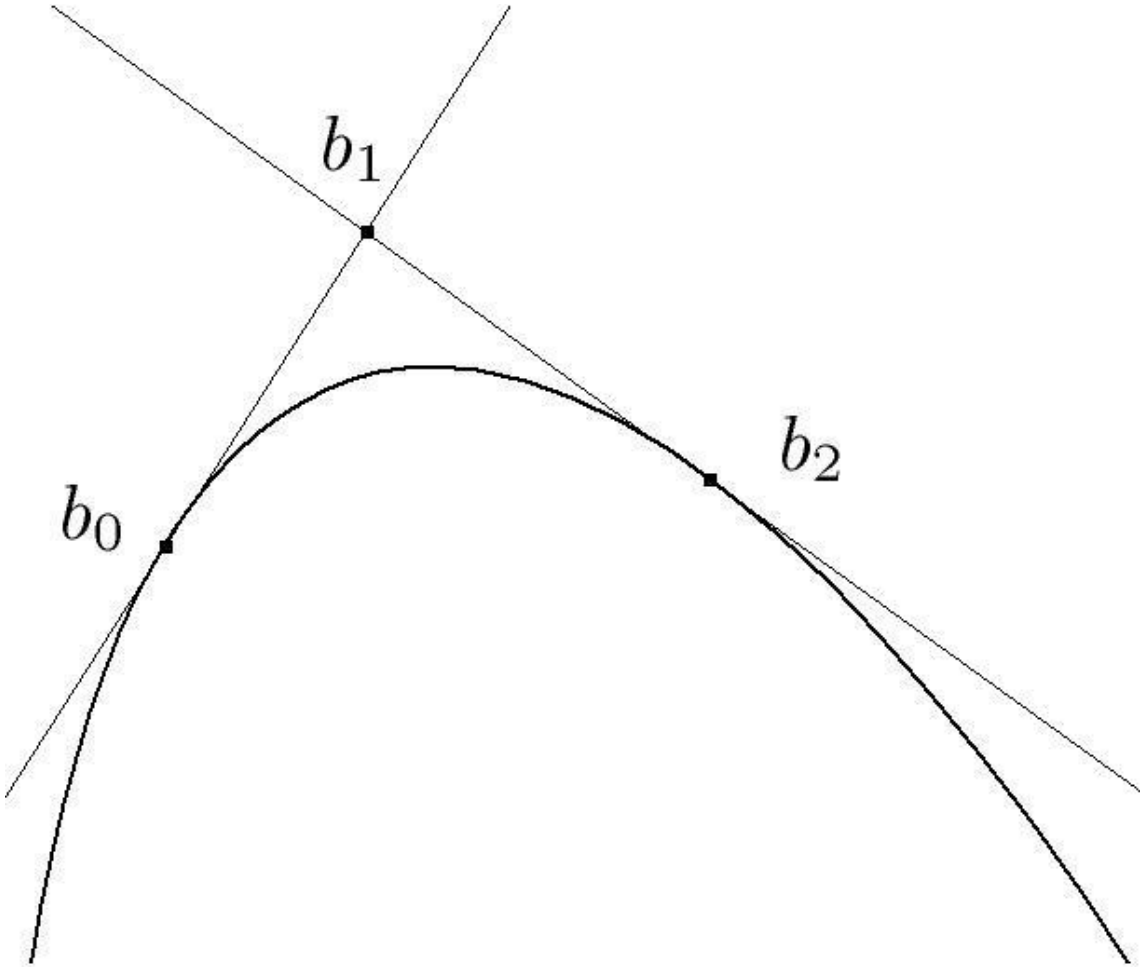
Sustituir picos por “Arcos Simples”.





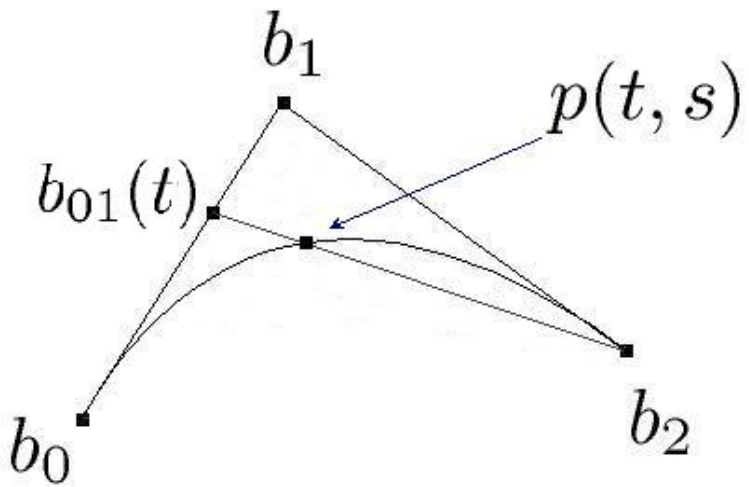
Construcción del Arco.

Definido por los puntos, $\mathbf{b}_0$, $\mathbf{b}_1$ y $\mathbf{b}_2$, **Puntos de Control**.



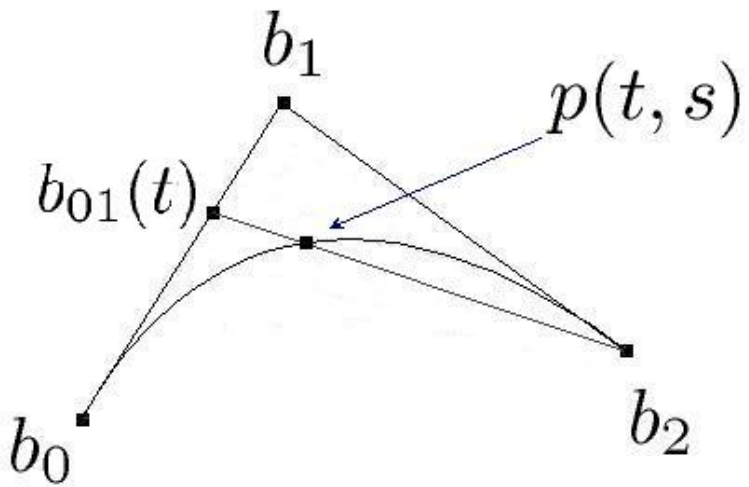
Parametrización del segmento $\mathbf{b}_0\mathbf{b}_1$

$$\mathbf{b}_{01}(t) = t(\mathbf{b}_1 - \mathbf{b}_0) + \mathbf{b}_0 \qquad t \in [0, 1]$$



Parametrización del segmento $\mathbf{b}_{01}(t)\mathbf{b}_2$

$$\mathbf{p}(t,s) = \mathbf{b}_{01}(t)s + (1-s)\mathbf{b}_2 \qquad s \in [0,1]$$



Representación Racional de Bézier

$$\mathbf{p}(t) = \frac{\delta_0 B_0^2(t)\mathbf{b}_0 + \delta_1 B_1^2(t)\mathbf{b}_1 + \delta_2 B_2^2(t)\mathbf{b}_2}{\delta_0 B_0^2(t) + \delta_1 B_1^2(t) + \delta_2 B_2^2(t)}$$

Donde los δ_i son llamados *pesos*,$t \in [0, 1]$ y

$$\delta_0 = -2\mathcal{C}(\mathbf{b}_2, \mathbf{b}_0),\ \delta_1 = -\mathcal{C}(\mathbf{b}_2, \mathbf{b}_0),\ \delta_2 = \mathcal{C}(\mathbf{b}_1)$$

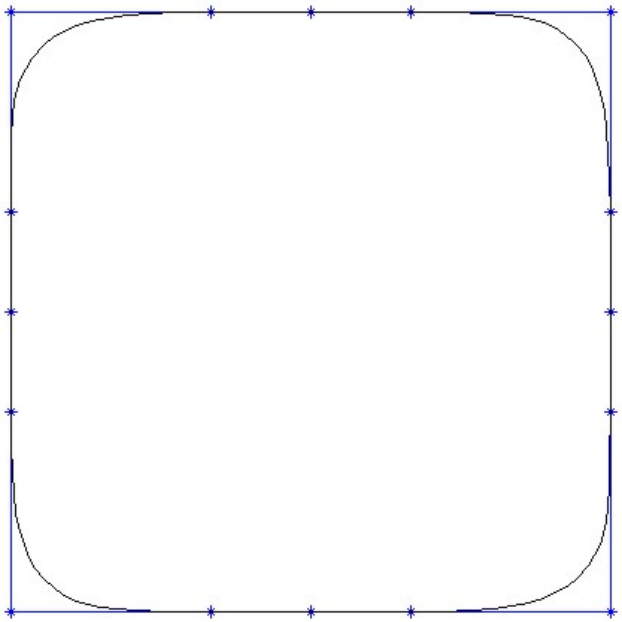
Teorema

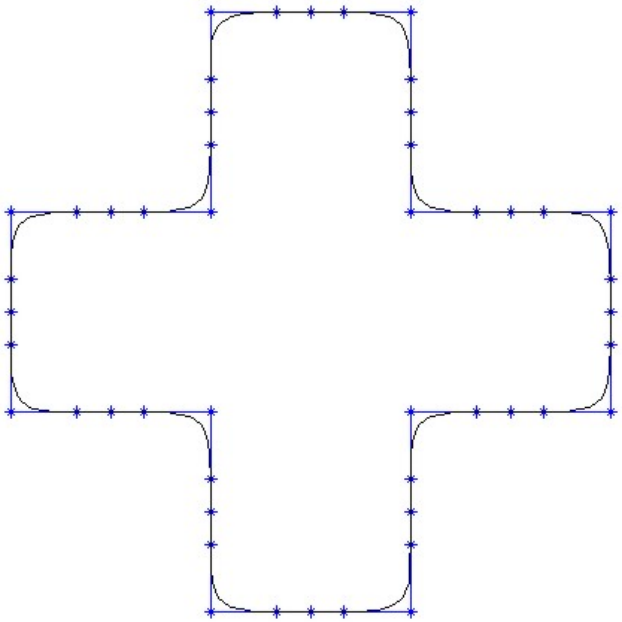
Todo **Arco Cónico** definido de $\mathbf{b}_0$ a $\mathbf{b}_2$ puede expresarse por medio de una **Curva Racional de Bézier** en la siguiente forma:

$$\mathbf{p}(t) = \frac{\delta_0 B_0^2(t)\mathbf{b}_0 + \delta_1 B_1^2(t)\mathbf{b}_1 + \delta_2 B_2^2(t)\mathbf{b}_2}{\delta_0 B_0^2(t) + \delta_1 B_1^2(t) + \delta_2 B_2^2(t)}, \qquad t \in [0,1].$$

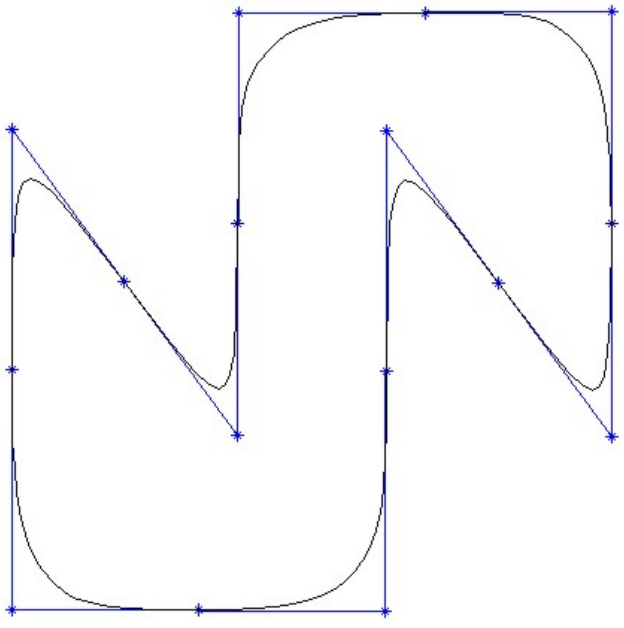
donde los δ_i son llamados **pesos** y $B_i^2(t)$ son los polinomios de **Bernstein**.

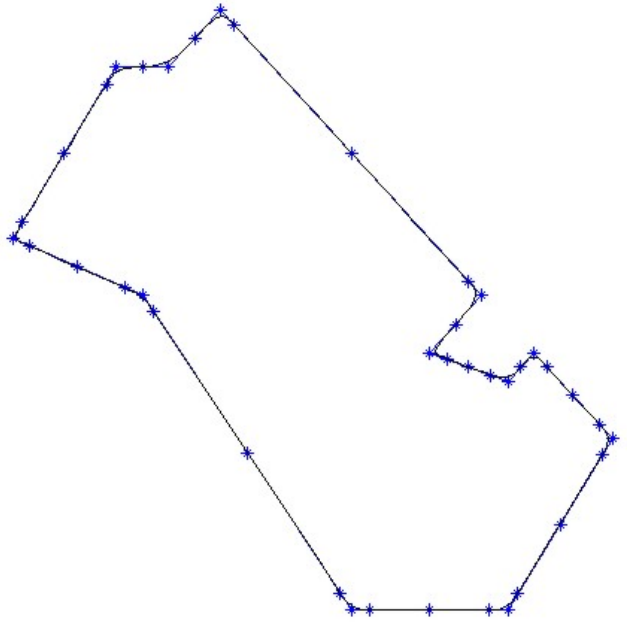
Puntos de Control. Lados Iguales





Puntos de Control. Longitudes Distintas



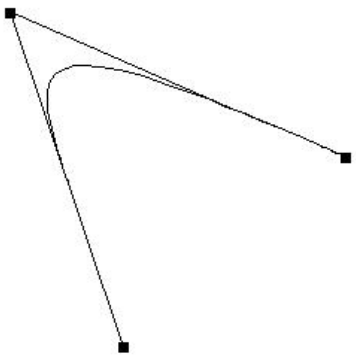


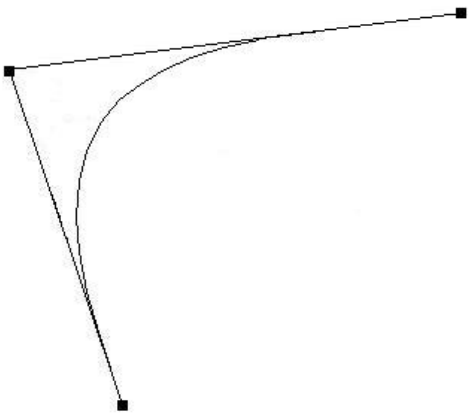
Pesos.

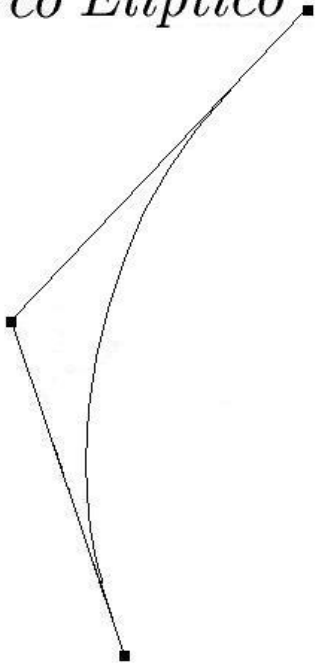
Si θ es el ángulo formado por los segmentos $\mathbf{b}_0\mathbf{b}_1$ y $\mathbf{b}_1\mathbf{b}_2$, entonces,

$0^o < \theta < 120^o$, $\theta = 120^o$, $120^o < \theta < 180^o$

Arco Hiperbólico *Arco Parabólico* *Arco Eliptico*







UNAMALLA en Matlab