
The Paraview logo features three vertical bars in red, green, and blue to the left of the word "ParaView" in a bold, sans-serif font. The word "Para" is in a lighter weight than "View".

Tutorial de **ParaView**

03a. Cómo entender los datos – El formato VTK

Michael Heredia Pérez
mherediap@unal.edu.co

Universidad Nacional de Colombia
Sede Manizales

El modelo VTK

To use ParaView effectively, you need to understand the ParaView data model. ParaView uses VTK, the Visualization Toolkit, to provide the visualization and data processing model. This chapter briefly introduces the VTK data model used by ParaView. For more details, refer to one of the VTK books.

Para un correcto uso del programa es necesario entender cómo funciona el software, no simplemente cuáles botones oprimir.

Data Object

Es el tipo de dato principal, puede ser tan sencillo como una grillas o tan complejo como árboles o grafos.

Se compone de dos elementos

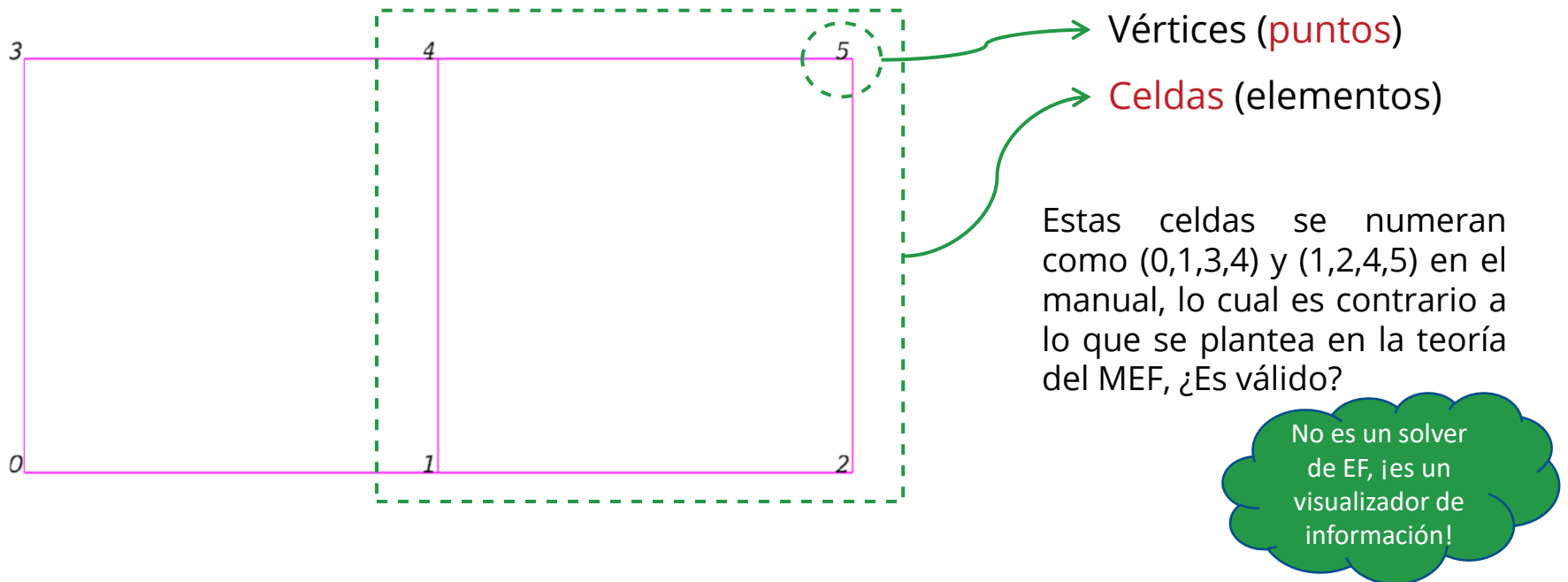
Mallas

Atributos

Topología

Geometría

Sobre las Mallas



- Siendo los puntos y las celdas las estructuras de visualización, la información estará localizada allí.
- Se habla de **conectividad** como el mapeo que hay desde las celdas a los vértices.
- Se definen por la topología y las coordenadas espaciales de los vértices.

Sobre los Atributos

Son **arreglos de información** que definen los valores discretos de un **campo** sobre la malla.

→ Velocidad, presión, esfuerzos, temperatura..... CUALQUIERA.

→ ¿Tienen forma? No, se asume dentro de ParaView según sus dimensiones.

Point-centered data



Atributos definidos en los puntos mediante interpolación según la documentación del formato VTK.

Cell-centered data

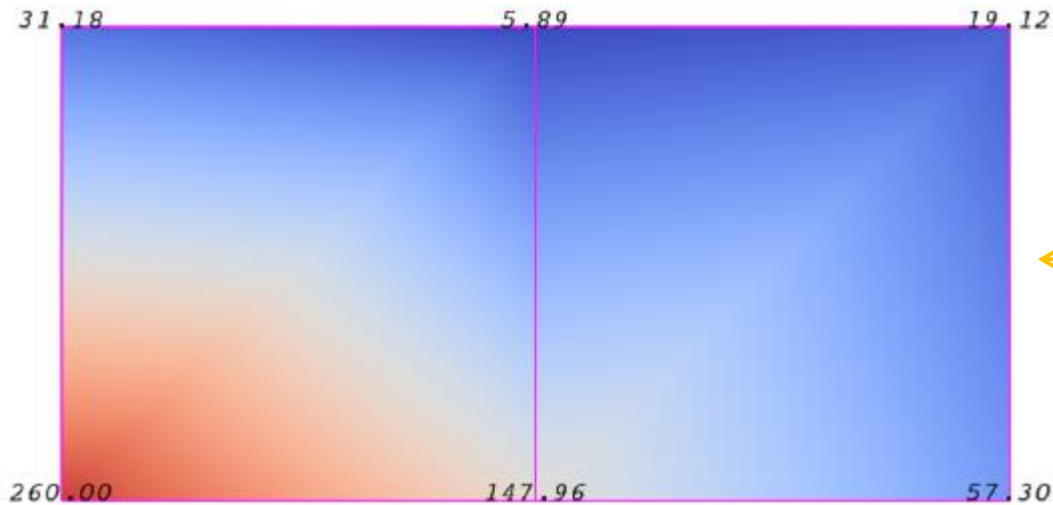


Atributos definidos en la celda constantes en ella.

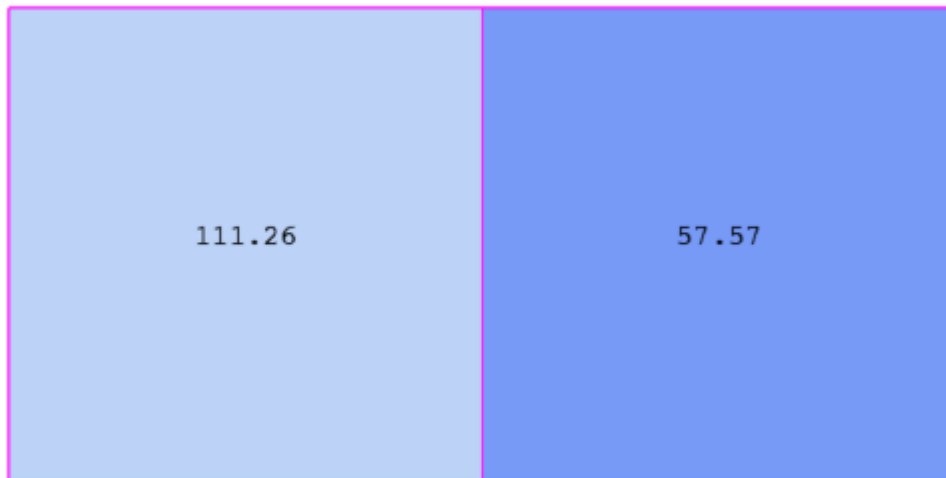
Se puede convertir de celda a punto mediante filtros

“Cell data to point data”
filter

Sobre los Atributos



← Point-centered data



← Cell-centered data

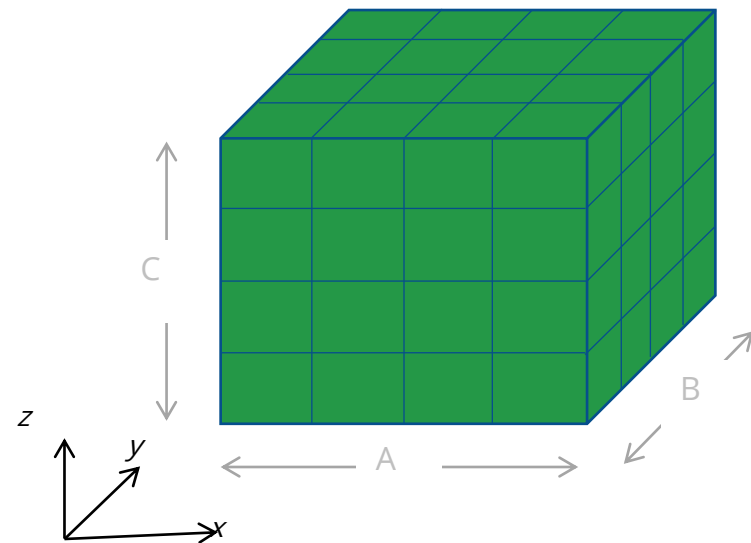
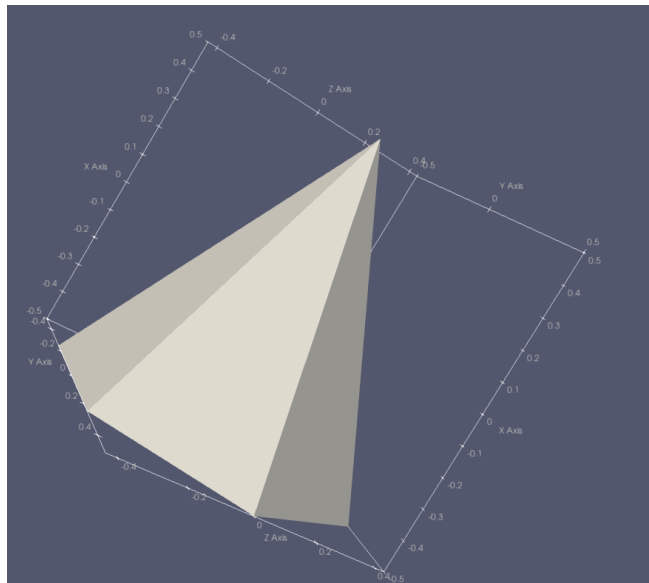
Algunos términos antes de abordar los objetos de datos

■ Tres arreglos:

Extents Límites en cada dirección, i.e. $(0, A_{n-1})(0, B_{n-1})(0, C_{n-1})$

Origin Posición del punto con índices $(0,0,0)$

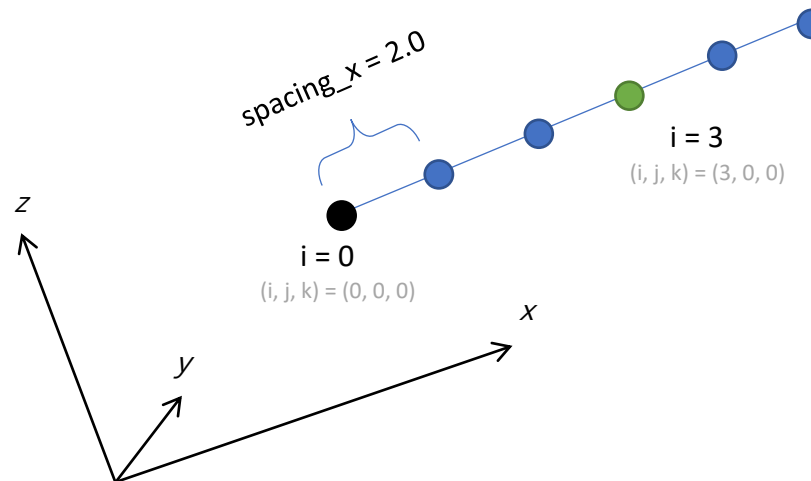
Spacing Distancia entre cada punto



Algunos términos antes de abordar los objetos de datos

Índices

Si determinada malla funciona de esta manera, los nodos estarán identificados por un índice (i, j, k) en el caso más general, el cual se empleará en la obtención de sus coordenadas.



Algunos términos antes de abordar los objetos de datos

Índice plano (*flat index*)

[Flat indexing: a compilation technique to enhance parallelism of logic programs - IEEE Conference Publication](#)

$$idx_flat = f(i, j, k)$$

Abstract:

The paper presents a systematic approach to the compilation of logic programs for efficient clause indexing. As the kernel of the approach, we propose the indexing tree which provides a simple, but precise representation of average parallelism per node (i.e., choice point) as well as the amount of clause trials. It also provides the way to evaluate the number of the cases that the control is passed to the failure code by the indexing instruction such as switch on term, switch on constant, or switch on structure. By analyzing the indexing tree created when using the indexing scheme implemented in the WAM, we show the drawback of the WAM indexing scheme in terms of parallelism exposition and scheduling. Subsequently we propose a new indexing scheme, which we call Flat indexing. Experimental results show that over one half of the benchmarks benefit from the Flat indexing, such that compared with the WAM indexing scheme, the number of choice points is reduced by 15%. Moreover, the amount of failures which occur during the execution of indexing instructions is reduced by 35%.

Published in: [Proceedings 1998 International Conference on Parallel and Distributed Systems \(Cat. No.98TB100250\)](#)

Date of Conference: 14-16 Dec. 1998

INSPEC Accession Number: 6135805

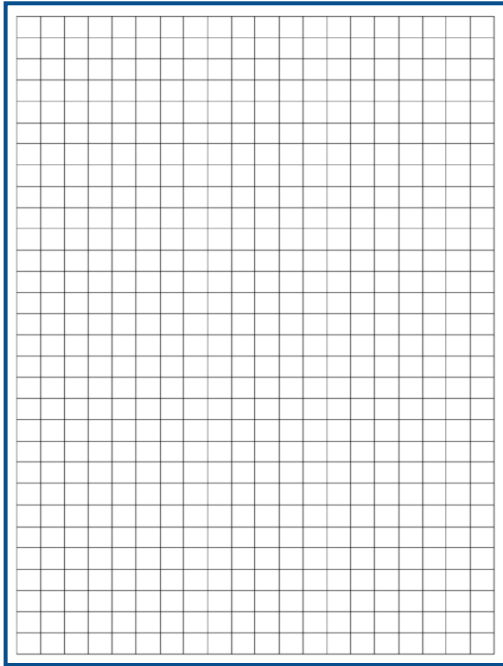
Date Added to IEEE Xplore: 06 August 2002

DOI: [10.1109/ICPADS.1998.741166](#)

Print ISBN: 0-8186-8603-0

Publisher: IEEE

Grilla rectilínea uniforme



Uniform Rectilinear Grid o *Image Data*.

Las más eficientes en almacenamiento y con algoritmos más optimizados.

Sus celdas son del mismo tipo, el cual es definido por la dimensión del conjunto de datos (*extents*), y puede ser:

- Vértices (0D)
- Líneas (1D)
- Píxeles o *pixels* (2D)
- Volúmenes o *voxels* (3D)

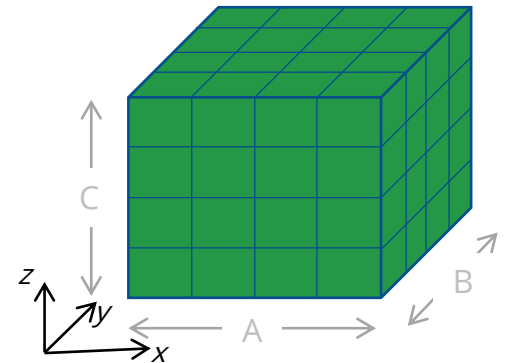
Topología y coordenadas implícitas.

Emplea:

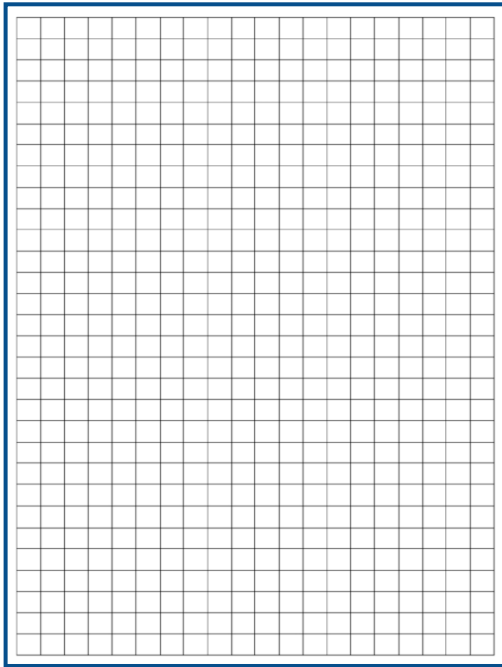
Extents Límites en cada dirección, i.e. $(0, A_{n-1})(0, B_{n-1})(0, C_{n-1})$

Origin Posición del punto con índices $(0,0,0)$

Spacing Distancia entre cada punto



Grilla rectilínea uniforme



Uniform Rectilinear Grid o *Image Data*.

Las coordenadas se de la forma:

$$coord = origen + índice * distancia$$

$$idx_flat = k * (npts_x * npts_y) + j * npts_x + i$$

El formato VTK permite el almacenamiento en índice plano.

Topología y coordenadas implícitas.

Emplea:

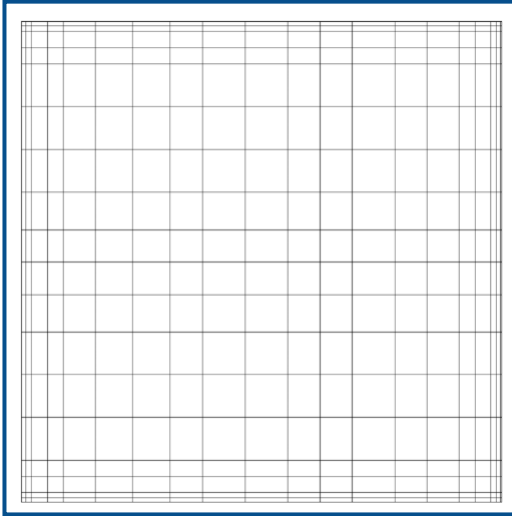
Extents Límites en cada dirección, i.e. $(0, A_{n-1})(0, B_{n-1})(0, C_{n-1})$

Origin Posición del punto con índices $(0, 0, 0)$

Spacing Distancia entre cada punto

Índice plano en función de los índices (i, j, k) .

Grilla rectilínea



Rectilinear Grid.



Topología implícita y coordenadas semi-implícitas.



Emplea:

- *Extents*
- Arrays de memoria en dirección x , y , z . para mejor ahorro computacional:

$$nptsx + nptsy + nptsz \neq nptsx * nptsy * nptsz$$



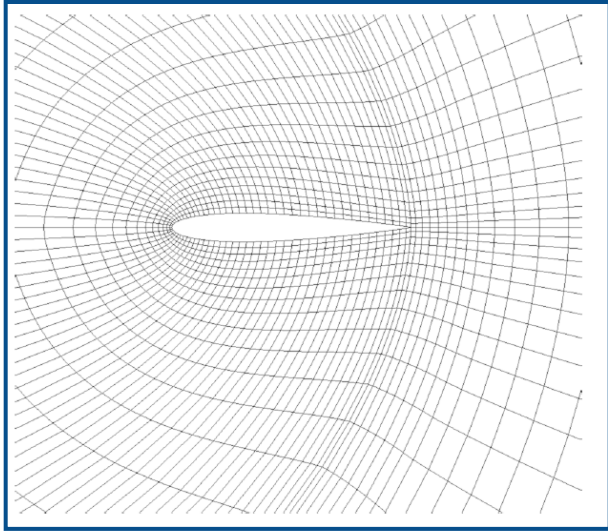
Coordenadas: $coord = (coord_array_x(i); coord_array_y(j); coord_array_z(k))$

$$idx_flat = k(npts_x * npts_y) + j * npts_x + i$$



Celdas del mismo tipo, determinando por las dimensiones del conjunto de datos (*extents*), desde 0D hasta 3D.

Grilla curvilinea



Curvilinear grid.



Topología implícita y coordenadas explícitas.



Emplea:

- *Extents*
- Arrays de puntos-coordenadas: posición de cada uno de los vertices de manera explícita



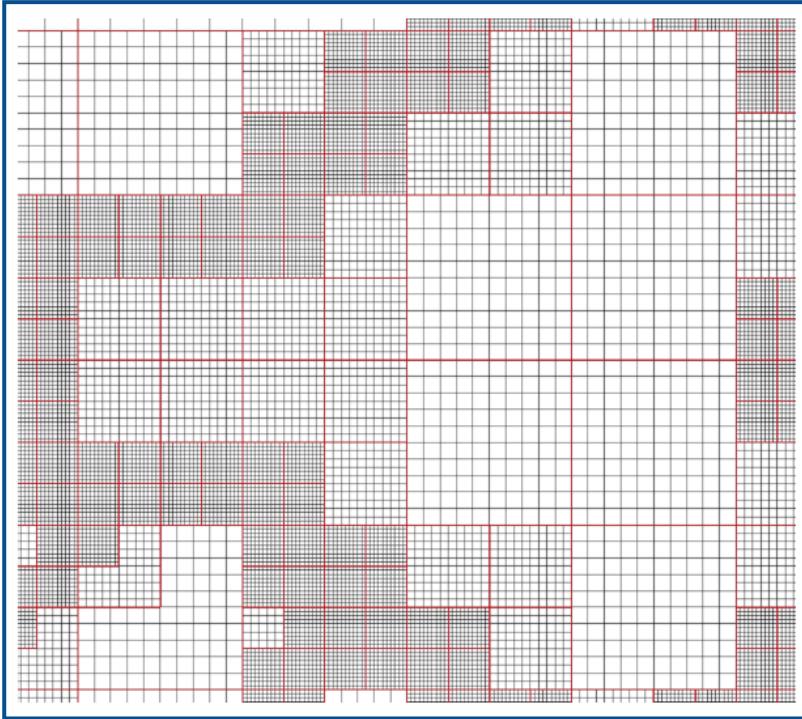
Coordenadas: $coord = vector_coord(idx_flat)$

$$idx_flat = k(npts_x * npts_y) + j*npts_x + i$$



Celdas del mismo tipo, determinando por las dimensiones del conjunto de datos (*extents*), desde 0D hasta 3D. 2D se llaman quad o cuádruples y 3D se llaman Hexaedros.

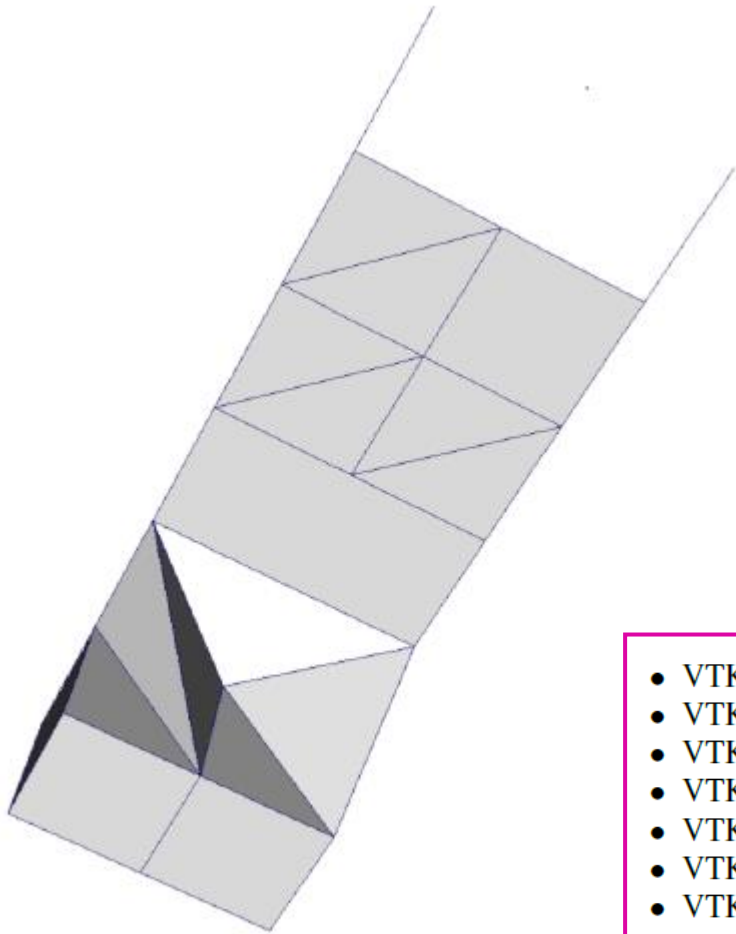
Conjunto de datos RMA



- ❖ *AMR dataset*: Adaptive Mesh Refinement.
- ❖ Colección de grillas rectangulares agrupadas según incrementos de coeficientes de refinamiento (*decreasing spacing*).
- ❖ Enmascara subregiones de la grilla rectilinear mediante arrays de bytes, lo cual permite que VTK pueda soportar el solapamiento (*overlapping*).

Es un conjunto
complejo de
datos

Grillas no estructuradas



Unstructured grids.

La topología y las coordenadas se almacenan de manera explícita.

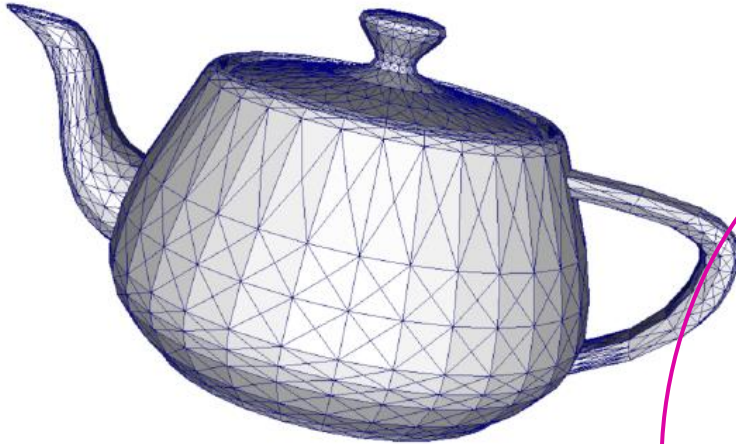
Altísimo consume de memoria.

Soporta diferentes tipos de celdas según la estructuración de VTK.

- VTK_EMPTY_CELL
- VTK_VERTEX
- VTK_POLY_VERTEX
- VTK_LINE
- VTK_POLY_LINE
- VTK_TRIANGLE
- VTK_TRIANGLE_STRIP

- VTK_POLYGON
- VTK_PIXEL
- VTK_QUAD
- VTK_TETRA
- VTK_VOXEL
- VTK_HEXAHEDRON
- VTK_WEDGE

Grillas poligonales



▀ *Polygonal grid o polydata.*

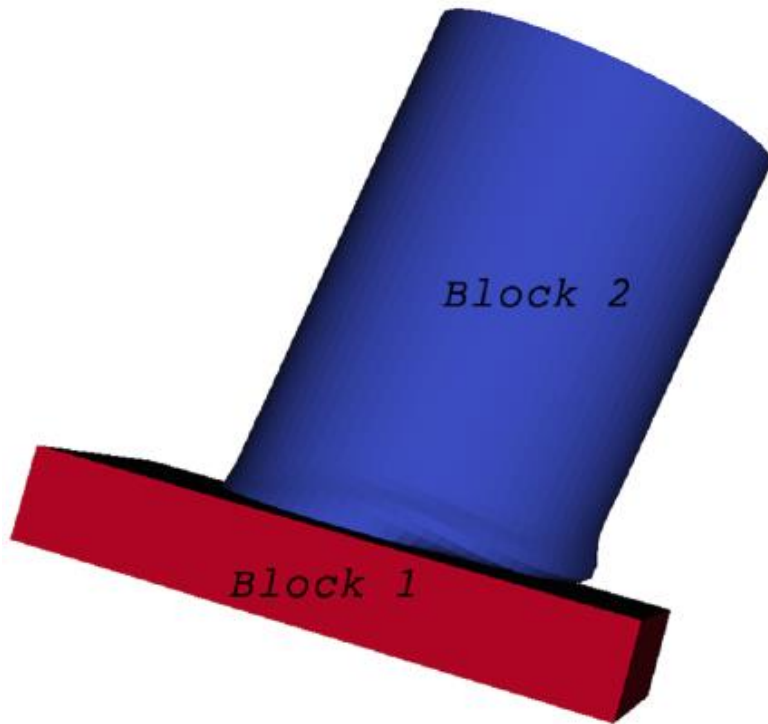
▀ Grillas no estructuras eficientes para el renderizado.

▀ Consiste de **celdas**:

- 0D: vertices y polivértices.
- 1D: líneas y polilíneas.
- 3D: polígonos y franjas triangulares

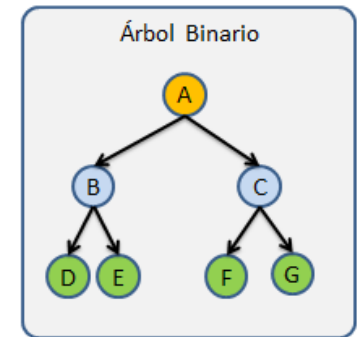
Hay filtros para convertir de grilla no estructurada a polidata y viceversa.

Multibloques



❖ *Multiblock dataset.*

❖ Árboles de memoria donde los nodos son conjuntos de datos simples.



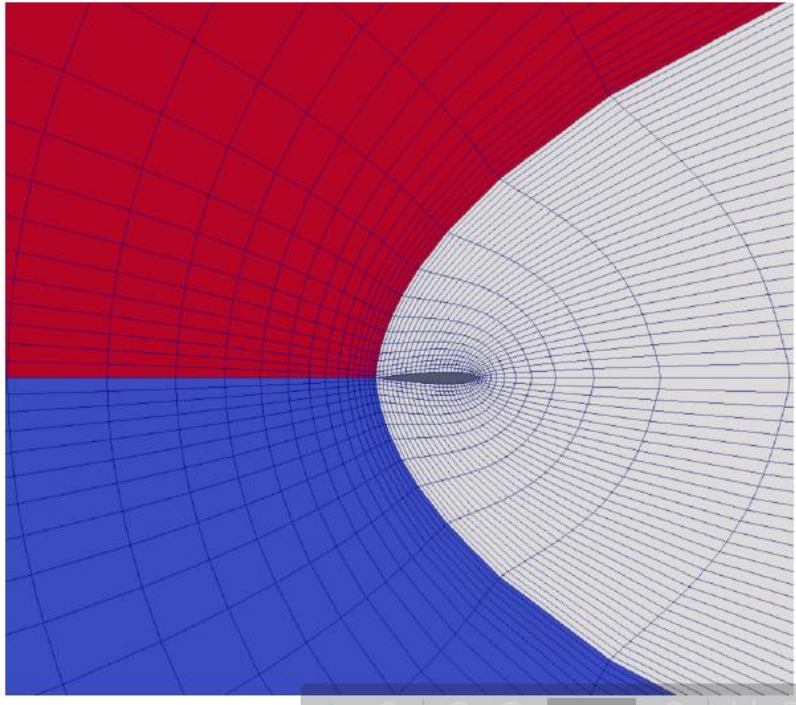
❖ Son conjuntos de datos relacionados ya sea dentro o fuera de ParaView.



❖ Los nodos pueden tener diferentes atributos entre sí.

❖ Es un conjunto de datos simples (todos except el *ARM*).

Conjuntos de datos multipieza



- ▣ *Multipiece dataset.*
- ▣ Agrupan conjuntos de datos simples que hacen parte de un mismo conjunto con los mismos atributos.
- ▣ No se puede crear internamente en ParaView.
- ▣ Se utilizan para recopilar conjuntos de datos producidos por una simulación paralela sin tener que unir las mallas.

Es un conjunto
simple de datos

Tablas

	Author	Affiliation			Alma Mater	Categories	Age	Coolness
0	Biff	NASA			Ole...	Jazz; Ro...	27	0.6
1	Bob	Bob's	Supermarket		Ole...	Jazz	54	0.3
2	Baz	Bob's	Supermarket		TVI	Food	16	0.3
3	Bippity	Oil	Changes	'R'	TVI	Food	23	0.2
4	Boppity	Oil	Changes	'R'	Home	Food; A...	34	0.25
5	Boo	Oil	Changes	'R'	Princeton	Automobiles	27	0.7

- ❖ *Tables.*
- ❖ Información en formato tabular compuesto de filas y columnas.
- ❖ Los filtros aplicables a las tablas generan tablas.
- ❖ Se pueden convertir a puntos o mallas estructuradas mediante filtros.
- ❖ Se pueden cargar con formato separación de coma .csv.

"Table to Point" y
"Table to Structured
Grid"