

Píndoles

Filtre JSXGraph – Aules

Ús i funcionalitats

1. Què és JSXGraph i per a què servix en Aules?

Què és JSXGraph?

JSXGraph és una biblioteca JavaScript de codi obert que permet crear construccions geomètriques interactives, traçar funcions matemàtiques, generar gràfics i visualitzar dades directament en el navegador, sense necessitat d'instal·lar cap programari adicional.

En el context d'Aules (Moodle), s'integra a través d'un filtre (plugin) que permet inserir estes construccions en qualsevol recurs de text: pàgines, etiquetes, preguntes de qüestionari, etc.

Nota tècnica

Implementació: JavaScript pur, sense dependències externes.

Compatibilitat: Funciona en tots els navegadors moderns (Chrome, Firefox, Safari, Edge).

Tipus de plugin: `filter_jsxgraph` — actua com a filtre de text, no com a mòdul d'activitat.

Per a quin tipus de continguts és útil JSXGraph en Aules?

JSXGraph resulta especialment útil en les següents situacions:

- Representació gràfica de funcions matemàtiques (àlgebra, càlcul, estadística).
- Construccions geomètriques interactives (punts, rectes, circumferències, polígons).
- Visualització de dades dinàmiques en activitats i recursos de curs.
- Preguntes de qüestionari amb resposta gràfica (combinat amb el tipus de pregunta STACK).
- Materials educatius interactius per a qualsevol àrea que requerisca representació visual.

Consell d'ús

Encara que el seu ús més habitual és en Matemàtiques i Física, JSXGraph pot emprar-se en qualsevol àrea on es necessite comunicar informació de manera gràfica i interactiva.

2. Ús i funcionament bàsic

Com s'inserix una construcció JSXGraph en un recurs d'Aules?

Per a inserir una construcció, seguix estos passos:

- **1.** Accedix a l'editor de text del recurs (pàgina, etiqueta, pregunta, etc.).
- **2.** Canvia al mode d'edició HTML (editor de codi font o editor de text pla).
- **3.** Inserix l'etiqueta `<jsxgraph>` amb el codi JavaScript de la construcció.
- **4.** Guarda el recurs. El filtre processarà l'etiqueta i renderitzarà la construcció.

Exemple:

```
<jsxgraph width="500" height="500" box="mybox">

JXG.Options.slider.snapValues = [-5, -2, -1, 0, 1, 2, 5];
JXG.Options.slider.snapValueDistance = 0.2;

const board = JXG.JSXGraph.initBoard(BOARDID, { boundingbox: [-10, 10, 10, -10],
axis: true });

var a = board.create('slider', [[2, -5], [7, -5], [-5, 1, 5]], { name: 'a' });
var b = board.create('slider', [[2, -6], [7, -6], [-5, 0, 5]], { name: 'b' });
var c = board.create('slider', [[2, -7], [7, -7], [-5, 0, 5]], { name: 'c' });

var f = board.create('functiongraph', [(x) => a.Value() * x * x + b.Value() * x +
c.Value()]);

var txt = board.create('text', [-9, -5,
() => JXG.Math.Numerics.generatePolynomialTerm([c.Value(), b.Value(), a.Value()],
2, 'x', 2)
], { fontSize: 18 });

</jsxgraph>
```

Nota tècnica

BOARDID: És una constant gestionada automàticament pel filtre. No és necessari definir-la manualment.

Editor recomanat: editor de text pla per a evitar que Moodle modifiqui el codi en guardar.

Quins atributs admet l'etiqueta <jsxgraph>?

L'etiqueta accepta els següents atributs principals:

Atribut	Valor per defecte	Descripció
<code>width</code>	500 (px)	Amplària del tauler en píxels.
<code>height</code>	400 (px)	Alçada del tauler en píxels.
<code>aspect-ràtio</code>	—	Ràtio amplària/altura. Si s'usa juntament amb <code>width</code> sense <code>height</code> , regula l'altura automàticament.
<code>numberOfBoards</code>	1	Permet usar múltiples taulers en un mateix bloc.
<code>box</code>	—	ANEU personalitzat per al contenidor. Si s'omet, es genera automàticament (BOARDID).
<code>ext_formules</code>	false	Activa l'extensió d'integració amb preguntes de tipus Fórmules.
<code>title</code>	—	Títol descriptiu del tauler (accessibilitat).
<code>description</code>	—	Descripció del tauler (accessibilitat).

És possible usar múltiples taulers en un mateix bloc?

Sí. Mitjançant l'atribut `numberOfBoards` es poden definir diversos taulers independents dins d'un mateix bloc. En eixe cas, els IDs de cada tauler s'accedixen mitjançant les constants BOARDID0, BOARDID1, BOARDID2, etc.

```
<jsxgraph width="500" height="300" numberOfBoards="2">
  var b1 = JXG.JSXGraph.initBoard(BOARDID0, { boundingbox: [-5,5,5,-5], axis: true
});
  var b2 = JXG.JSXGraph.initBoard(BOARDID1, { boundingbox: [-3,3,3,-3], axis: true
});
</jsxgraph>
```

Nota tècnica

Quan s'usen múltiples taulers, els atributs `width`, `height` i `description` poden rebre valors separats per espais, un per tauler.

3. Integració amb altres elements d'Aules. JSXGraph + STACK

Pot JSXGraph usar-se dins de preguntes de qüestionari?

Sí. JSXGraph es pot utilitzar en dos contextos dins dels qüestionaris:

a) Com a element visual estàtic (il·lustració): Inserint el bloc `<jsxgraph>` en l'enunciat de qualsevol tipus de pregunta per a mostrar una construcció com a suport visual.

b) Com a element interactiu amb resposta gràfica (Preguntes STACK): Combinant JSXGraph amb el tipus de pregunta STACK. Això permet que l'alumne interactue amb la gràfica (per exemple, arrossegant un punt) i que la resposta quedi registrada en Moodle.

JSXGraph + STACK

JSXGraph és una biblioteca de visualització interactiva, per sí mateixa no avalua respostes matemàtiques. STACK és un tipus de pregunta de Moodle que integra el sistema d'àlgebra computacional Maxima per a avaluar respostes matemàtiques de manera simbòlica.

STACK permet incrustar JSXGraph dins de la pregunta i a més **enllaçar els valors del gràfic amb la resposta de l'estudiant**, per exemple:

- L'estudiant mou un punt en el gràfic → eixa coordenada s'envia com a resposta → Maxima l'avalua.

Això dona lloc a preguntes on la resposta és la interacció gràfica, una cosa impossible amb JSXGraph a soles.

Característica	Descripció
JSXGraph a soles	Visualització interactiva. No avalua respostes ni assigna puntuació.
STACK	Avaluació simbòlica amb Maxima. Feedback automàtic personalitzat.
STACK + JSXGraph	El millor dels dos: gràfica interactiva + avaluació matemàtica real.
Aleatorització	STACK genera paràmetres aleatoris amb Maxima per a cada estudiant.
Dificultat	JSXGraph és més senzill. STACK requereix aprendre sintaxi Maxima.

Recomanació: Usa JSXGraph només per a contingut exploratori sense avaluació. Usa STACK quan necessites qualificar automàticament.

Exemple 1: Pendent d'una recta

L'estudiant veu una recta aleatòria i ha d'escriure la seua pendent numèricament.

Variables de la pregunta

```
m: rand_with_prohib(-5, 5, [0]); /* pendent aleatòria, mai 0 */
b: rand_with_prohib(-4, 4, [0]); /* intercepte aleatori */
```

Enunciat de la pregunta (fragment clau)

```
<p>Observa la següent recta i determina la seua
<strong>pendent</strong>.</p>
```

```

<div style="display:flex; justify-content:center;">
[[jsxgraph width="400px" height="400px"]]
var board = JXG.JSXGraph.initBoard(BOARDID, {
  boundingbox: [-8, 8, 8, -8],
  axis: true,
  showCopyright: false
});

var m = {#m#};
var b = {#b#};

board.create('functiongraph', [function(x){ return m*x + b; }], {
  strokeColor: '#c00',
  strokeWidth: 2
});

board.create('point', [0, b], {
  name:'(0, '+b+')',
  fixed: true,
  color: 'blue'
});

board.create('point', [1, m+b], {
  name:'(1, +(m+b)+)',
  fixed: true,
  color: 'blue'
});
[[/jsxgraph]]
</div>

<p>Escriu el pendent de la recta: [[input:ans1]]
[[validation:ans1]]</p>

```

Nota: {#m#} injecta el valor de Maxima en JavaScript. {@m@} ho mostra en el text del feedback .

Configuración de l'Entrada: ans1

Camp	Valor
Tipus d'entrada	Algebraica
Resposta model	m
Prohibir flotants	Sí

Arbre de respostes potencials: prt1 - Node 1

Camp	Valor
Prova	AlgEquiv
TAns	M
Si la prova és verdadera	1 Retroalimentació: Correcte! El pendent és {@m@}.
Si la prova és falsa	0 Retroalimentació: Recorda que el pendent és el coeficient que acompanya a x . Intenta de nou.

¿Cuál es la pendiente de la recta? Versión 3 (última)

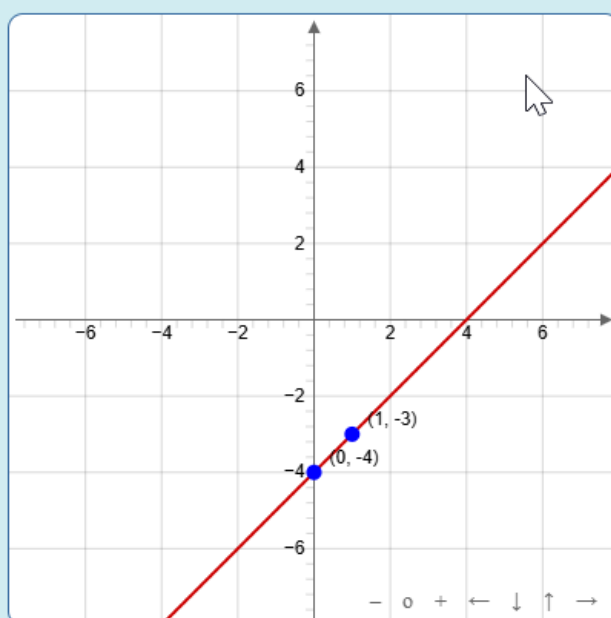
Pregunta 1

Sin responder aún

Se puntúa como 0 sobre 1,00

Observa la siguiente recta y determina su **pendiente**.

Question is missing tests or variants.



Escribe la pendiente de la recta:

Exemple 2: Arrossegar el punt a l'arrel d'una paràbola

L'estudiant arrossega un punt sobre l'eix X fins on creu que està l'arrel. JSXGraph envia el valor automàticament a STACK sense que l'estudiant haja d'escriure res.

Variables de la pregunta

```
a: 1;
r1: rand_with_prohib(-4, 4, [0]); /* arrel aleatòria */
b: -r1; /* i = x^2 - r1*x → arrel en x=r1 */
c: 0;
```

Enunciat de la pregunta (Patró clau)

Este és el patró fonamental per a enviar valors gràfics a STACK:

<p>La paràbola que veus té una arrel en $x = 0$ i una altra arrel. Arrossega el punt taronja fins on creus que està la segona arrel.</p>

```
[[jsxgraph input-ref-ans1="ans1Ref" width="500px" height="400px"]]

var board = JXG.JSXGraph.initBoard(BOARDID, {
  boundingbox: [-6, 8, 8, -4],
  axis: true,
  showCopyright: false
});

var a = {#a#};
var b = {#b#};
var c = {#c#};

board.create('functiongraph', [function(x){
  return a*x*x + b*x + c;
}], { strokeColor: '#7c6fff', strokeWidth: 2.5 });

var p = board.create('point', [1, 0], {
  name: 'La teua resposta',
  color: '#e8a020',
  size: 6,
  snapToGrid: true,
  snapSizeX: 1,
  snapSizeY: 1
});

p.I = function(){ return 0; };
p.update();

/* Escoltar l'arrossegament directament sobre el punt */
p.on('drag', function(){
  var val = Math.round(p.X());
  var input = document.getElementById(ans1Ref);
  if(input){
    input.value = val;
    /* Disparar esdeveniment de canvi perquè STACK detecte el nou valor */
    input.dispatchEvent(new Event('change'));
    input.dispatchEvent(new Event('input'));
  }
});

[[/jsxgraph]]

<p>[[input:ans1]] [[validation:ans1]]</p>
```

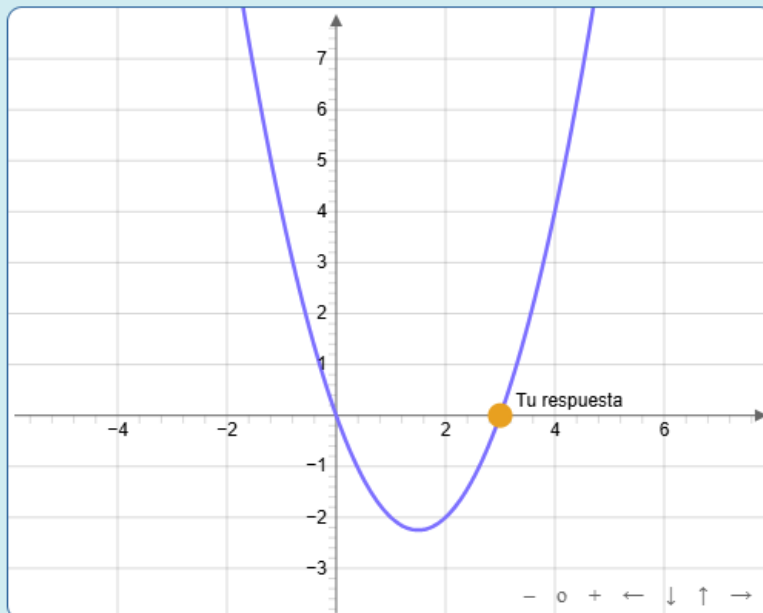
Arrastra el punto hasta la raíz de la parábola Versión 6 (última)

Pregunta 1

Sin responder aún

Se puntúa como 0 sobre 1,00

La parábola que ves tiene **una raíz en $x = 0$** y otra ¹ Question is missing tests or variants. raíz. Arrastra el punto **naranja** hasta donde crees que está la segunda raíz.



3

Tu respuesta fue interpretado como:

3

Per què este patró i no uns altres?:

- **input-ref-ans1="ans1Ref"** en l'etiqueta: fa que STACK resolga automàticament l'ID real de l'input, que canvia amb cada versió de la pregunta.
- **p.on('drag')** en lloc de `board.on('update')`: escolta directament l'arrossegament del punt, que és més fiable.
- **dispatchEvent('change')** i **dispatchEvent('input')**: notifiquen a STACK del nou valor perquè el processe correctament.
- Mai usar l'ID hardcodedo (ej: q965578:1_ans1): eixe ID canvia amb cada pregunta i versió.

Configuració de l'Entrada: ans1

Camp	Valor
Tipus d'entrada	Numerical
Resposta model	r1
Extra options	mindp:0, maxdp:0

Arbre de respostes potencials: prt1 - Node 1

Camp	Valor
Prova	NumAbsolute
TAns	r1
Tolerancia	0.6
Si la prova és verdadera	1 Correcte! L'arrel estava en $x = \{@r1@\}$.
Si la prova és falsa	0 No és correcte. L'arrel estava en $x = \{@r1@\}$.

4. Referència ràpida de sintaxi

Injecció de valors Maxima en JSXGraph

Sintaxi	Ús
<code>{#variable#}</code>	Injecta el valor numèric de Maxima en el codi JavaScript de JSXGraph.
<code>{@variable@}</code>	Mostra el valor calculat de Maxima en el text HTML (enunciat o feedback).

Funcions Maxima més útils en STACK

Funció	Descripció
<code>rand_with_prohib(min, max, [exclousos])</code>	Enter aleatori excloent valors no desitjats.
<code>rand(n)</code>	Enter aleatori entre 0 i n-1.
<code>diff(f, x)</code>	Derivada de f respecte a x.
<code>solve([ec], [var])</code>	Resol una equació.
<code>ev(expr, x=valor)</code>	Avalua una expressió en un punt.

Tests de resposta PRT més comuns

Test	Quan usar-lo
AlgEquiv	Respostes algebraiques: comprova equivalència simbòlica (ej: $x+1 = 1+x$).
NumAbsolute	Respostes numèriques amb tolerància absoluta (ej: 3 amb tolerància 0.5).
NumRelative	Respostes numèriques amb tolerància relativa (percentatge).

Test	Quan usar-lo
<code>EqualComAss</code>	Igualtat commutativa i associativa (útil per a sumes i productes).

5. Problemes freqüents i solucions

La construcció no es mostra després de guardar el recurs. Què pot estar passant?

Este és el problema més habitual. Les causes més freqüents són:

- **Editor incorrecte:** L'editor de text enriquit de Moodle (TinyMCE) pot eliminar o modificar l'etiqueta `<jsxgraph>` en guardar. Solució: canviar a l'editor de text pla abans d'inserir el codi, entrar al camp 'Enunciat de la pregunta' i canviar a Format HTML.
- **Filtre inactiu:** El filtre JSXGraph pot no estar habilitat.
- **Etiqueta malformada:** Comprovar que l'etiqueta d'obertura i tancament són correctes: `<jsxgraph> ... </jsxgraph>`.

Nota tècnica

Si tens configurat TinyMCE com a editor per defecte i necessites inserir codi JSXGraph, canvia temporalment a l'editor de text pla en les teues preferències d'usuari (Panell de control → Preferències → Preferències de l'editor).

Per què el tauler es mostra però apareix en blanc (sense contingut)?

Si el contenidor (div) es renderitza, però la construcció està buida, les causes habituals són:

- **Error de JavaScript:** Obrir la consola del navegador (F12 → Consola) per a detectar errors de sintaxis en el codi.
- **BOARDID incorrecte:** Si s'usa un ID personalitzat mitjançant l'atribut `box`, assegurar-se de referenciar-lo correctament en el codi JS.
- **Conflicte de IDs:** Si hi ha diversos taulers en la mateixa pàgina amb el mateix ID, només s'inicialitzarà el primer. Usar `numberOfBoards` o deixar que el filtre genere els IDs automàticament.

Nota tècnica

Depuració ràpida: La consola del navegador és la ferramenta principal. Qualsevol error en el codi JSXGraph apareixerà ací amb la seua línia exacta.

Com es gestiona la responsivitat del tauler en diferents dispositius?

Perquè el tauler s'adapte a la grandària de pantalla, és recomanable usar l'atribut `aspect-ratio` en lloc de definir `width` i `height` fixos:

```
<jsxgraph width="100%" aspect-ràtio="16/9">
  var board = JXG.JSXGraph.initBoard(BOARDID, {
    boundingbox: [-5, 5, 5, -5],
    axis: true
  });
</jsxgraph>
```

Si el que es vol és centrar el gràfic simplement cal encapsular el codi JSXGraph en un contenidor (div):

```
<div style="display:flex; justify-content:center;">
<jsxgraph width="100%" aspect-ràtio="16/9">
  var board = JXG.JSXGraph.initBoard(BOARDID, {
    boundingbox: [-5, 5, 5, -5],
    axis: true
  });
</jsxgraph>
</div>
```

Nota tècnica

Regla: Si s'especifiquen tant `width` com `height`, l'atribut `aspect-ràtio` s'ignora.

Recomanació: Per a contextos on l'alumne pugui accedir des de mòbil o tauleta, definir `width` en percentatge i controlar l'altura mitjançant `aspect-ràtio` proporciona una experiència més consistent.

Error d'unitats CSS

Error: The width/height of a JSXGraph must use a known CSS-length unit.

Causa: l'etiqueta `[[jsxgraph]]` no inclou la unitat en l'ample o alt.

Solució: afegir `px` explícitament:

```
[[jsxgraph width="400px" height="400px"]]
```

Error de versió de Maxima

Error: The version of the STACK-Maxima libraries does not match.


Causa: el plugin STACK va ser actualitzat però les biblioteques de Maxima no. Requereix intervenció d'administrador.

**El codi JSXGraph funciona en pàgines, però no en preguntes de qüestionari.
Per què?**

Este és un comportament esperat. La previsualització durant l'edició d'una pregunta no renderitza el codi JSXGraph, però sí que funciona correctament quan:

- Guardes la pregunta i la previsualitzes des del banc de preguntes.
- Afiges la pregunta a un qüestionari i el previsualitzes com a estudiant.
- Els estudiants realitzen el qüestionari.

Passos per a verificar que el codi funciona:

1. Edita la pregunta i inserix el codi JSXGraph en l'enunciat (usant editor de text pla).
2. Guarda els canvis.
3. Vés al banc de preguntes del curs.
4. Localitza la pregunta i fes clic en la icona de "Vista prèvia" (ull .
5. La construcció JSXGraph es mostrarà correctament en la finestra emergent.

He inserit codi JavaScript amb operadors de comparació (<, >) i JSXGraph no funciona. Què està passant?

El problema més habitual és que Moodle interpreta els símbols < i > com a etiquetes HTML i pot modificar-los o eliminar-los en guardar. Això trenca el codi JavaScript.

Exemple de codi problemàtic:


```
<jsxgraph width="500px" height="400px">
  var board = JXG.JSXGraph.initBoard(BOARDID, {boundingbox: [-5,5,5,-5]});
  for (var k = 0; k < 5; k++) { // ✘ El símbol < causa problemes
    board.create('point', [k, k*k]);
  }
</jsxgraph>
```

Solució: Substituir els operadors '<' i '>' per les seues entitats HTML equivalents:

- < → <
- > → >
- <= → ≤
- >= → ≥

Exemple de codi corregit:

```
<jsxgraph width="500px" height="400px">
  var board = JXG.JSXGraph.initBoard(BOARDID, {boundingbox: [-5,5,5,-5]});
```

```
for (var k = 0; k &lt; 5; k++) { //  Usa &lt; en lloc de <  
board.create('point', [k, k*k]);  
}  
</jsxgraph>
```

Nota tècnica

Altres entitats HTML útils en codi JavaScript:

- & → &
- " → "
- ' → ' (encara que generalment no és necessari)

6. Recursos i referències

On puc trobar exemples i documentació addicional?

Els recursos oficials i més útils per a treballar amb JSXGraph en Aules són:

- **Repositori oficial del plugin (GitHub):** https://github.com/jsxgraph/moodle-filter_jsxgraph
- **Biblioteca d'exemples exportables:** <https://jsxgraph.org/share> — permet exportar directament al format del filtre Moodle.
- **Documentació de la API de JSXGraph:** <https://jsxgraph.org/docs/>
- **Integració amb preguntes Fórmules:** https://github.com/jsxgraph/moodleformulas_jsxgraph

Consell de productivitat

Els exemples de jsxgraph.org/share inclouen l'opció d'exportar directament en format d'etiqueta <jsxgraph> llista per a pegar en Moodle, la qual cosa permet reutilitzar construccions sense necessitat d'escriure el codi des de zero.