

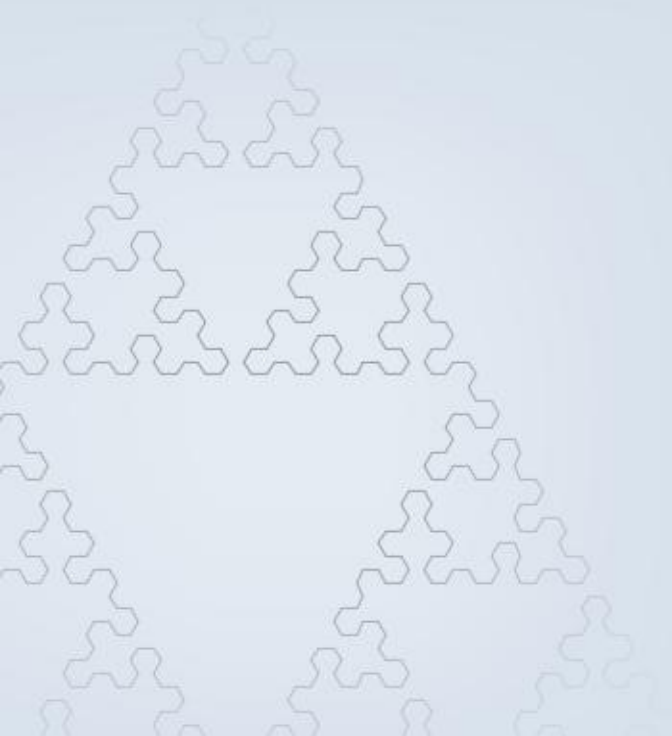
Dynamische Geometrie mit JSXGraph

Matthias Ehmann, Michael Gerhäuser, Carsten Miller,
Bianca Valentin, Alfred Wassermann, Peter Wilfahrt

Universität Bayreuth

```
var brd = JSXGraph.initBoard('box', {ax: 10, ay: 10, bx: 400, by: 400});
var s = brd.createElement('slider', [[1, 3], [5, 5]]);
var a = brd.createElement('slider', [[1, 2], [5, 5]]);
var f = brd.createElement('slider', [[1, 1], [5, 5]]);
function(x) { return Math.sin(x); }
plot = brd.createElement('functiongraph', {f: f, s: s, a: a});
os = brd.createElement('riemannsum', [f, function() { return s.Value(); }, function() { return a.Value(); }, function() { return b.Value(); }], {fillColor: '#ffff00'});
```

Zukunftstrends in dynamischer Geometrie



```
var brd = JSXGraph.initBoard('box', {axis: true});
var s = brd.createElement('slider', [[1, 3], [5, 5]]);
var a = brd.createElement('slider', [[1, 2], [5, 5]]);
var b = brd.createElement('slider', [[1, 1], [5, 5]]);
var f = function(x) { return Math.sin(x); };
var plot = brd.createElement('functiongraph', {
  f: f,
  x0: 0,
  x1: 2 * Math.PI,
  y0: -1,
  y1: 1,
  fillColor: '#ffff00',
  strokeColor: 'black',
  strokeDash: [5, 5],
  strokeWidth: 2,
  style: 'solid'
});
```

Zukunftstrends in dynamischer Geometrie

Verwenden von Dateien, die mit einem DGS wie z.B. GEONE_xT, GeoGebra oder Cinderella erstellt wurden

```
var brd = JSXGraph.initBoard('box', {ax: 1, ay: 1, width: 400, height: 400});  
var s = brd.createElement('slider', [[1, 3], [5, 5]]);  
var a = brd.createElement('slider', [[1, 2], [5, 5]]);  
var b = brd.createElement('slider', [[1, 1], [5, 5]]);  
var f = function(x) { return Math.sin(x); };  
var plot = brd.createElement('functiongraph', {f: f, x: 0, x2: 2, y: -1, y2: 1});  
var os = brd.createElement('riemannsum', [f, function() { return s.Value(); }, function() { return a.Value(); }, function() { return b.Value(); }], {fillColor: '#ffff00'});
```

Zukunftstrends in dynamischer Geometrie

Verwenden von Dateien, die mit einem DGS wie z.B. GEONE_xT, GeoGebra oder Cinderella erstellt wurden

und

Erstellen von geometrischen Konstruktionen sowie
Veranschaulichen von Graphen

```
brd = JSXGraph.initBoard('box', {axi  
s = brd.createElement('slider', [[1, 3], [5  
brd.createElement('slider', [[1, 2], [5  
b = brd.createElement('slider', [[1, 1], [5  
f = function(x) { return Math.sin(x); }  
plot = brd.createElement('functiongraph',  
os = brd.createElement('riemannsum', [f,  
function() { return s.Value(); }, function()  
function() { return a.Value(); },  
function() { return b.Value(); }  
fillColor: '#ffff00';
```

Zukunftstrends in dynamischer Geometrie

Verwenden von Dateien, die mit einem DGS wie z.B. GEONE_xT, GeoGebra oder Cinderella erstellt wurden

und

Erstellen von geometrischen Konstruktionen sowie

Veranschaulichen von Graphen

OHNE Plugins wie Java oder

Flash zu benutzen!

Warum nicht Java oder Flash?

```
var brd = JXG.JSXGraph.initBoard('box', {axis: true});
var s = brd.createElement('slider', [[1, 3], [5, 5]]);
var a = brd.createElement('slider', [[1, 2], [5, 5]]);
var b = brd.createElement('slider', [[1, 1], [5, 5]]);
var f = function(x) { return Math.sin(x); };
var plot = brd.createElement('functiongraph', {
  f: f,
  s: s,
  a: a,
  b: b,
  fillColor: '#ffff00'
});
```

Warum nicht Java oder Flash?

```
var brd = JXG.JSXGraph.initBoard('box', {axs:  
  s = brd.createElement('slider', [[1,3],[5  
  a = brd.createElement('slider', [[1,2],[5  
  b = brd.createElement('slider', [[1,1],[5  
  f = function(x) { return Math.sin(x); }  
  plot = brd.createElement('functiongraph',  
  os = brd.createElement('riemannsum', [f,  
  function() { return s.Value(); }, function()  
  function() { return a.Value(); },  
  function() { return b.Value(); }  
  ,  
  fillColor: '#ffff00',
```


Warum nicht Java oder Flash?

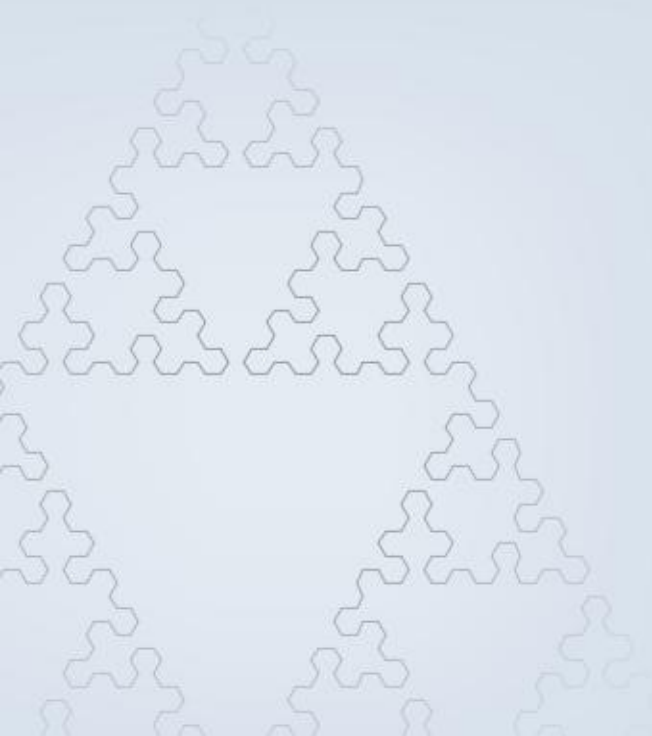
- benötigen Installation externer Plugins
- lange Initialisierungszeit
- Probleme mit Rechnern ohne Windows

```
var board = JXG.JSXGraph.initBoard('box', {axis: true});
var s = brd.createElement('slider', [[1, 3], [5, 5]]);
var a = brd.createElement('slider', [[1, 2], [5, 5]]);
var b = brd.createElement('slider', [[1, 1], [5, 5]]);
var f = function(x) { return Math.sin(x); };
var plot = brd.createElement('functiongraph', {
  f: f,
  color: '#ff00ff',
  fillColor: '#ffff00'
});
```

Warum nicht Java oder Flash?

- benötigen Installation externer Plugins
- lange Initialisierungszeit
- Probleme mit Rechnern ohne Windows
- Wie soll man neue Geräte wie iPhone, iPad, Blackberry, Palm Pre, Android, Google Chrome OS verwenden?

JSXGraph



```
var brd = JSXGraph.initBoard('box', {axis: 2});  
var s = brd.createElement('slider', [[1, 3], [5, 5]]);  
var a = brd.createElement('slider', [[1, 2], [5, 5]]);  
var b = brd.createElement('slider', [[1, 1], [5, 5]]);  
var f = function(x) { return Math.sin(x); };  
var plot = brd.createElement('functiongraph', {  
  f: f,  
  s: s,  
  a: a,  
  b: b,  
  fillColor: '#ffff00',  
});
```

JSXGraph

- open source Projekt bei sourceforge.net

```
var brd = JSXGraph.initBoard('box', {axis: [1, 2]});  
var s = brd.createElement('slider', [[1, 3], [5, 5]]);  
var a = brd.createElement('slider', [[1, 2], [5, 5]]);  
var b = brd.createElement('slider', [[1, 1], [5, 5]]);  
var f = function(x) { return Math.sin(x); };  
var plot = brd.createElement('functiongraph', {  
  f: f,  
  os = brd.createElement('riemannsum', [f, function() { return s.Value(); }], function() { return a.Value(); }], function() { return b.Value(); }],  
  fillColor: '#ffff00',  
});
```

JSXGraph

- open source Projekt bei sourceforge.net
- entwickelt an der Universität Bayreuth

```
var brd = JSXGraph.initBoard('box', {ax: 1, ay: 1, bx: 5, by: 5});  
var s = brd.createElement('slider', [[1, 3], [5, 5]]);  
var a = brd.createElement('slider', [[1, 2], [5, 5]]);  
var b = brd.createElement('slider', [[1, 1], [5, 5]]);  
var f = function(x) { return Math.sin(x); };  
var plot = brd.createElement('functiongraph', {f: f});  
var os = brd.createElement('riemannsum', [f, function() { return s.Value(); }, function() { return a.Value(); }, function() { return b.Value(); }], {fillColor: '#ffff00'});
```

JSXGraph

- open source Projekt bei sourceforge.net
- entwickelt an der Universität Bayreuth
- komplett in JavaScript implementiert

```
var brd = JSXGraph.initBoard('box', {ax: 1, ay: 1, bx: 5, by: 5});  
var s = brd.createElement('slider', [[1, 3], [5, 5]]);  
var a = brd.createElement('slider', [[1, 2], [5, 5]]);  
var b = brd.createElement('slider', [[1, 1], [5, 5]]);  
var f = function(x) { return Math.sin(x); };  
var plot = brd.createElement('functiongraph', {f: f});  
var os = brd.createElement('riemannsum', [f, function() { return s.Value(); }, function() { return a.Value(); }, function() { return b.Value(); }], {fillColor: '#ffff00'});
```

JSXGraph

- open source Projekt bei sourceforge.net
- entwickelt an der Universität Bayreuth
- komplett in JavaScript implementiert
- kein Plugin nötig

```
var brd = JSXGraph.initBoard('box', {ax1:  
  s = brd.createElement('slider', [[1,3], [5  
  a = brd.createElement('slider', [[1,2], [5  
  b = brd.createElement('slider', [[1,1], [5  
  f = function(x) { return Math.sin(x); }  
  plot = brd.createElement('functiongraph',  
  os = brd.createElement('riemannsum', [f,  
  function() { return s.Value(); }, function()  
  function() {return a.Value();},  
  function() {return b.Value();}  
  fillColor: '#ffff00';
```


JSXGraph

- open source Projekt bei sourceforge.net
- entwickelt an der Universität Bayreuth
- komplett in JavaScript implementiert
- kein Plugin nötig
- unterstützt alle gängigen Broser (incl. Internet Explorer)

```
var brd = JSXGraph.initBoard('box', {ax  
s = brd.createElement('slider', [[1, 3], [5  
a = brd.createElement('slider', [[1, 2], [5  
b = brd.createElement('slider', [[1, 1], [5  
f = function(x) { return Math.sin(x); }  
g = brd.createElement('functiongraph',  
os = brd.createElement('riemannsum', [f,  
function() { return s.Value(); }, function()  
function() {return a.Value();},  
function() {return b.Value();}  
fillColor: '#ffff00';
```

JSXGraph

- open source Projekt bei sourceforge.net
- entwickelt an der Universität Bayreuth
- komplett in JavaScript implementiert
- kein Plugin nötig
- unterstützt alle gängigen Broser (incl. Internet Explorer)
- keine Initialisierungszeit

```
var brd = JSXGraph.initBoard('box', {axs:  
  s = brd.createElement('slider', [[1,3],[5  
  a = brd.createElement('slider', [[1,2],[5  
  b = brd.createElement('slider', [[1,1],[5  
  f = function(x){ return Math.sin(x); }  
  g = brd.createElement('functiongraph',  
  os = brd.createElement('riemannsum', [f,  
function(){ return s.Value();}, function()  
function(){return a.Value();},  
function(){return b.Value();}  
fillColor:'#ffff00;
```

Möglichkeiten von JSXGraph

```
var brd = JSXGraph.initBoard('box', {axis: true});  
var s = brd.createElement('slider', [[1, 3], [5, 5]]);  
var a = brd.createElement('slider', [[1, 2], [5, 5]]);  
var b = brd.createElement('slider', [[1, 1], [5, 5]]);  
var f = function(x) { return Math.sin(x); };  
var plot = brd.createElement('functiongraph', {  
  f: f,  
  s: s,  
  a: a,  
  b: b,  
  fillColor: '#ffff00',  
});
```

Möglichkeiten von JSXGraph

- **Dynamische Geometrie:**

Euklidische and homogene Koordinaten, Linien, Kreise, Polygone, Gleiter, Animationen, Kegelschnitte, Schieberegler,...

```
var brd = JSXGraph.initBoard('box', {ax: 1, ay: 1, width: 300, height: 300});
var s = brd.createElement('slider', [[1, 3], [5, 5]]);
var a = brd.createElement('slider', [[1, 2], [5, 5]]);
var b = brd.createElement('slider', [[1, 1], [5, 5]]);
var f = function(x) { return Math.sin(x); };
var plot = brd.createElement('functiongraph', {
  f: f,
  x: 0,
  x2: 2,
  y: -1,
  y2: 1,
  fillColor: '#ffff00',
});
var os = brd.createElement('riemannsum', [f,
function() { return s.Value(); }, function()
function() {return a.Value();},
function() {return b.Value();}
]);
```

Möglichkeiten von JSXGraph

- **Dynamische Geometrie:**

Euklidische and homogene Koordinaten, Linien, Kreise, Polygone, Gleiter, Animationen, Kegelschnitte, Schieberegler,...

- **Kurven:**

Funktionsgraphen, parametrisierte Kurven, Datenplots, Bezierkurven

```
brd = JSXGraph.initBoard('box', {axs:  
s = brd.createElement('slider', [[1, 3], [5  
a = brd.createElement('slider', [[1, 2], [5  
b = brd.createElement('slider', [[1, 1], [5  
f = function(x) { return Math.sin(x); }  
plot = brd.createElement('functiongraph',  
os = brd.createElement('riemannsum', [f,  
function() { return s.Value(); }, function()  
function() { return a.Value(); },  
function() { return b.Value(); }  
fillColor: '#ffff00';
```

Möglichkeiten von JSXGraph

- **Dynamische Geometrie:**

Euklidische and homogene Koordinaten, Linien, Kreise, Polygone, Gleiter, Animationen, Kegelschnitte, Schieberegler,...

- **Kurven:**

Funktionsgraphen, parametrisierte Kurven, Datenplots, Bezierkurven

- **Turtle Grafiken**

```
brd = JSXGraph.initBoard('box', {axs  
s = brd.createElement('slider', [[1, 3], [5  
a = brd.createElement('slider', [[1, 2], [5  
b = brd.createElement('slider', [[1, 1], [5  
f = function(x) { return Math.sin(x); }  
plot = brd.createElement('functiongraph',  
os = brd.createElement('riemannsum', [f,  
function() { return s.Value(); }, function()  
function() { return a.Value(); },  
function() { return b.Value(); }  
fillColor: '#ffff00';
```

Möglichkeiten von JSXGraph

- **Dynamische Geometrie:**

Euklidische and homogene Koordinaten, Linien, Kreise, Polygone, Gleiter, Animationen, Kegelschnitte, Schieberegler,...

- **Kurven:**

Funktionsgraphen, parametrisierte Kurven, Datenplots, Bezierkurven

- **Turtle Grafiken**

- **Diagramme**

```
brd = JSXGraph.initBoard('box', {axs:  
s = brd.createElement('slider', [[1, 3], [5  
a = brd.createElement('slider', [[1, 2], [5  
b = brd.createElement('slider', [[1, 1], [5  
f = function(x) { return Math.sin(x); }  
plot = brd.createElement('functiongraph',  
os = brd.createElement('riemannsum', [f,  
function() { return s.Value(); }, function()  
function() { return a.Value(); },  
function() { return b.Value(); }  
fillColor: '#ffff00';
```

Möglichkeiten von JSXGraph

- **Dynamische Geometrie:**

Euklidische and homogene Koordinaten, Linien, Kreise, Polygone, Gleiter, Animationen, Kegelschnitte, Schieberegler,...

- **Kurven:**

Funktionsgraphen, parametrisierte Kurven, Datenplots, Bezierkurven

- **Turtle Grafiken**

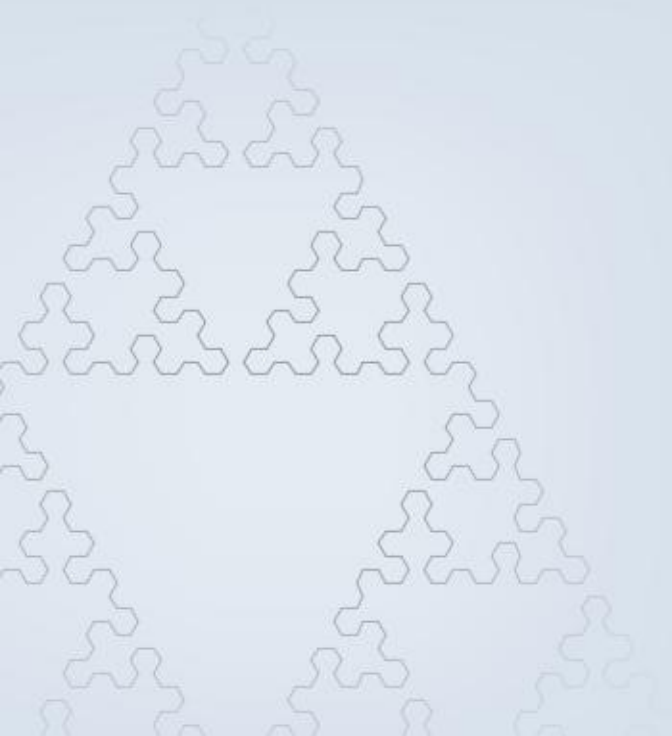
- **Diagramme**

- **Dateien anzeigen:**

GEONE_xT, Geogebra, Cinderella (zum Teil), Intergeo, ArcView (Karten)

```
brd = JSXGraph.initBoard('box', {axs  
s = brd.createElement('slider', [[1,3], [5  
a = brd.createElement('slider', [[1,2], [5  
b = brd.createElement('slider', [[1,1], [5  
f = function(x) { return Math.sin(x); }  
plot = brd.createElement('functiongraph',  
os = brd.createElement('riemannsum', [f,  
function() {return s.value();}, function()  
function() {return a.value();},  
function() {return b.value();}  
fillColor: '#ffff00';
```


Erstes Beispiel

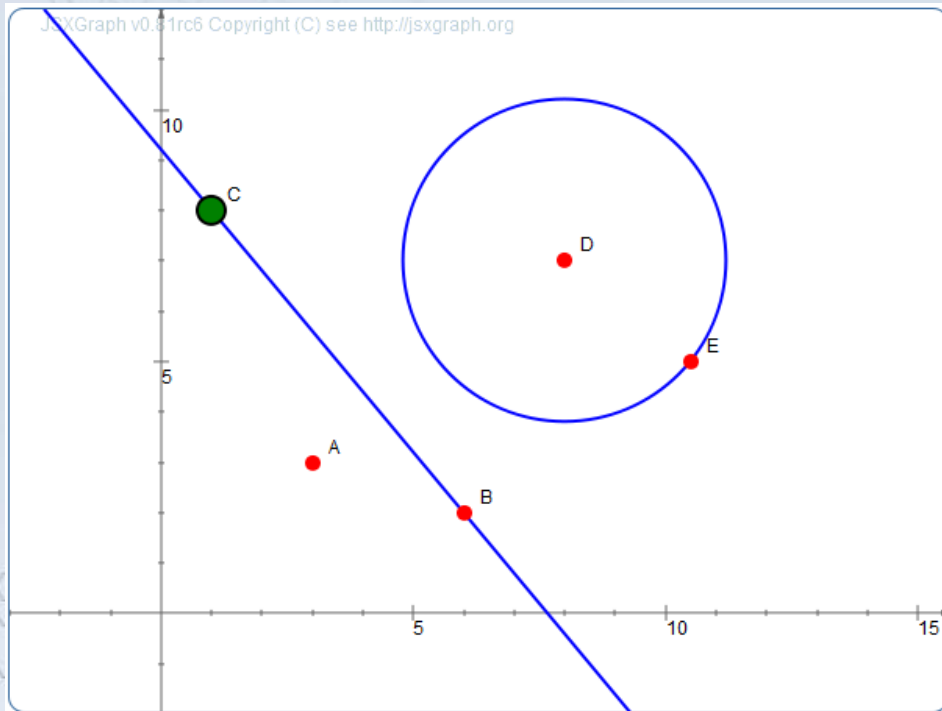


```
var brd = JSXGraph.initBoard('box', {ax: 1, ay: 1, bx: 5, by: 5});
var s = brd.createElement('slider', [[1, 3], [5, 5]]);
var a = brd.createElement('slider', [[1, 2], [5, 5]]);
var b = brd.createElement('slider', [[1, 1], [5, 5]]);
var f = function(x) { return Math.sin(x); };
var plot = brd.createElement('functiongraph', {
  f: f,
  x: 0,
  x2: 2,
  y: -1,
  y2: 1,
  fillColor: '#ffff00',
  strokeColor: 'black',
  strokeDash: [5, 5],
  strokeWidth: 2,
  title: 'f(x) = sin(x)'
});
```

Erstes Beispiel

```
var brd = JXG.JSXGraph.initBoard('box',
    {boundingbox:[-3, 12, 12, -2],
      keepaspectratio:true, shownavigation:false,
      showcopyright:false,axis:true});
var a = brd.create('point', [3,3], {name:'A'});
var b = brd.create('point', [6,2], {name:'B'});
var c = brd.create('point', [1,8],
    {name:'C', strokeColor:'black', fillColor:'green',
      highlightFillColor:'black', size:8});
var l = brd.create('line', [b, c], {name:'g'});
var d = brd.create('point', [8,7], {name:'D'});
var e = brd.create('point', [10.5,5], {name:'E'});
var cr = brd.create('circle', [d, e], {name:'k_1'});
var i1 = brd.create('intersection', [cr, l, 0],
    {face:'square'});
var i2 = brd.create('intersection', [cr, l, 1],
    {face:'', size:7});
```

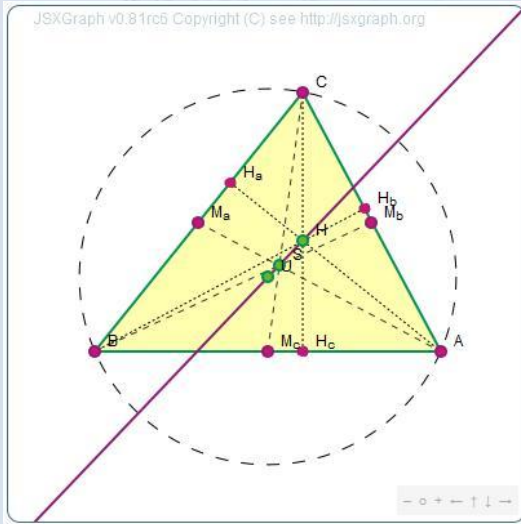
Erstes Beispiel



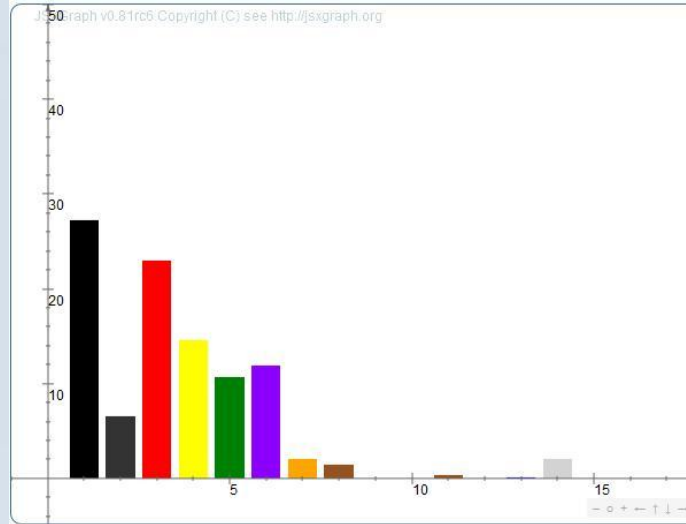
[Live Version](#)

```
var a = brd.create('point', [3,3],  
{name:'A'});  
var b = brd.create('point', [6,2],  
{name:'B'});  
var c = brd.create('point', [1,8],  
{name:'C',strokeColor:'black',  
fillColor:'green',size:8});  
var l = brd.create('line',[b, c]);  
var d = brd.create('point', [8,7],  
{name:'D'});  
var e = brd.create('point',  
[10.5,5], {name:'E'});  
var cr = brd.create('circle',  
[d, e], {name:'k_1'});  
var i1 =  
brd.create('intersection',  
[cr, l, 0], {face:'square'});  
var i2 =  
brd.create('intersection',  
[cr, l, 1], {face:'[]', size:7});  
os = brd.createElement('functiongraph',  
function(){ return s.Value();}, function()  
function(){return a.Value();},  
function(){return b.Value();}  
fillColor:'#ffff00');
```

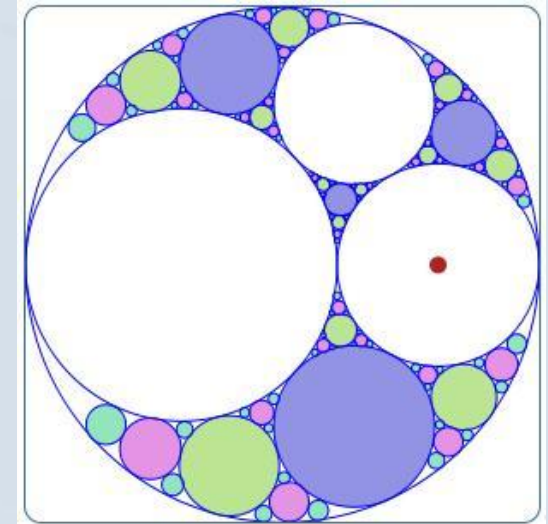
Weitere Beispiele



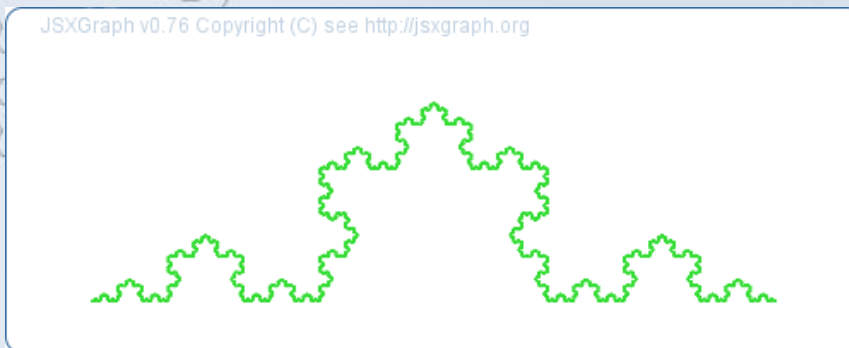
Geometrie: Eulergerade



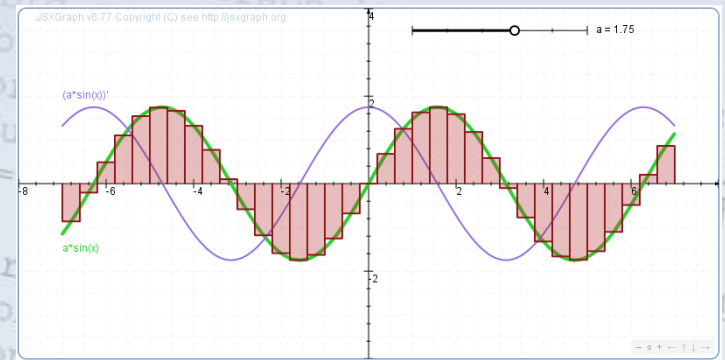
Interaktives Balkendiagramm



Apollonische Kreispackung



Koch Kurve mit einer Turtle



Funktionsdiskussion

Konstruieren mit JessieScript

```
var brd = JXG.JSXGraph.initBoard('box', {axs:  
  s = brd.createElement('slider', [[1,3],[5  
  a = brd.createElement('slider', [[1,2],[5  
  b = brd.createElement('slider', [[1,1],[5  
  f = function(x) { return Math.sin(x); }  
  plot = brd.createElement('functiongraph',  
  os = brd.createElement('riemannsum', [f,  
  function() { return s.Value(); }, function()  
  function() { return a.Value(); },  
  function() { return b.Value(); }  
  ,  
  fillColor: '#ffff00',
```

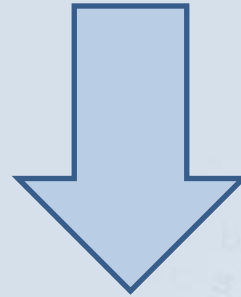
Konstruieren mit JessieScript

Die Programmierung mit JavaScript, um mit JSXGraph Geometrie am Rechner anzeigen zu können, stellt eine große Hürde dar.

```
var brd = JSXGraph.initBoard('box', {ax: 1, ay: 1, bx: 10, by: 10});
var s = brd.createElement('slider', [[1, 3], [5, 5]]);
var a = brd.createElement('slider', [[1, 2], [5, 5]]);
var b = brd.createElement('slider', [[1, 1], [5, 5]]);
var f = function(x) { return Math.sin(x); };
var plot = brd.createElement('functiongraph', {
  f: f,
  x: 0,
  x2: 10,
  y: 0,
  y2: 10,
  fillColor: '#ffff00',
});
var os = brd.createElement('riemannsum', [f, function() { return s.Value(); }, function() { return a.Value(); }, function() { return b.Value(); }]);
```

Konstruieren mit JessieScript

Die Programmierung mit JavaScript, um mit JSXGraph Geometrie am Rechner anzeigen zu können, stellt eine große Hürde dar.



Eine Alternativlösung ist nötig!

```
brd = JSXGraph.initBoard('box', {ax  
s = brd.createElement('slider', [[1,3], [5  
a = brd.createElement('slider', [[1,2], [5  
b = brd.createElement('slider', [[1,1], [5  
f = function(x) { return Math.sin(x); }  
os = brd.createElement('functiongraph',  
function() { return s.Value();}, function()  
function() {return a.Value();},  
function() {return b.Value();}  
fillColor: '#ffff00';
```

Konstruieren mit JessieScript

JSXGraph kann eine Syntax parsen, die sehr ähnlich zu dem ist, was in den Schulen gelehrt wird.

```
var brd = JSXGraph.initBoard('box', {ax: 10, ay: 10, width: 400, height: 400});
var s = brd.createElement('slider', [[1, 3], [5, 5]]);
var a = brd.createElement('slider', [[1, 2], [5, 5]]);
var b = brd.createElement('slider', [[1, 1], [5, 5]]);
var f = function(x) { return Math.sin(x); };
var plot = brd.createElement('functiongraph', {
  f: f,
  x: 0,
  x2: 2,
  y: -1,
  y2: 1,
  fillColor: '#ffff00',
});
```


Konstruieren mit JessieScript

JSXGraph kann eine Syntax parsen, die sehr ähnlich zu dem ist, was in den Schulen gelehrt wird, z.B.

$P(1,1)$

$Q(-2,2)$

$g=[PQ]$

```
var brd = JSXGraph.initBoard('box', {ax: 1, ay: 1, width: 300, height: 300});
var s = brd.createElement('slider', [[1, 3], [5, 5]]);
var a = brd.createElement('slider', [[1, 2], [5, 5]]);
var b = brd.createElement('slider', [[1, 1], [5, 5]]);
var f = function(x) { return Math.sin(x); };
var plot = brd.createElement('functiongraph', {f: f, fillColor: '#ffff00', strokeColor: 'black', strokeDash: [5, 5]});
var os = brd.createElement('riemannsum', [f, a, b], {fillColor: '#ffff00', strokeColor: 'black', strokeDash: [5, 5]});
function() { return s.Value(); }, function() { return a.Value(); }, function() { return b.Value(); }
```

Konstruieren mit JessieScript

JSXGraph kann eine Syntax parsen, die sehr ähnlich zu dem ist, was in den Schulen gelehrt wird, z.B.

$P(1,1)$

$Q(-2,2)$

$g=[PQ]$

$k(Q,1.5)$

$|| (g,R)$

```
var board = JSXGraph.initBoard('box', {axis: true});
var s = brd.createElement('slider', [[1,3], [5,5]]);
var a = brd.createElement('slider', [[1,2], [5,5]]);
var b = brd.createElement('slider', [[1,1], [5,5]]);
var f = function(x) { return Math.sin(x); };
var plot = brd.createElement('functiongraph', {
  f: f,
  fillColor: '#ffff00',
});
var riemannsum = brd.createElement('riemannsum', [f, s, a, b], function() {
  return s.Value();
}, function() {
  return a.Value();
}, function() {
  return b.Value();
});
```

Konstruieren mit JessieScript

JSXGraph kann eine Syntax parsen, die sehr ähnlich zu dem ist, was in den Schulen gelehrt wird, z.B.

$P(1,1)$

$Q(-2,2)$

$g=[PQ]$

$k(Q,1.5)$

$|| (g,R)$



einfach zu verwenden!

```
var board = JSXGraph.initBoard('box', {axis: true});
var s = brd.createElement('slider', [[1,3],[5,5]]);
var f = brd.createElement('slider', [[1,2],[5,5]]);
var g = brd.createElement('slider', [[1,1],[5,5]]);
var plot = brd.createElement('functiongraph', {
  f: function(x) { return Math.sin(x); }
});
var riemannsum = brd.createElement('riemannsum', [f,
  function() { return s.Value(); }, function() {
    return g.Value(); }
  ], {fillColor: '#ffff00'});
```

Beispiele

$|(P,g)$

$g=]AB[$

$P(1,1)$

$1/2(A,B)$

$\langle(A,B,C)$

$Y[A,B,C,D]$

Konstruieren mit JessieScript

$Q(g,3,2)$

$f:x^2+2*x+5$

$|_-(P,g)$

$X=g&k1$


$k(A,[PQ])$

```
var brd = JSXGraph.initBoard('box', {axis: true});
var s = brd.createElement('slider', [[1, 3], [5, 5]]);
var a = brd.createElement('slider', [[1, 2], [5, 5]]);
var b = brd.createElement('slider', [[1, 1], [5, 5]]);
var f = function(x) { return Math.sin(x); };
var plot = brd.createElement('functiongraph', {
  f: f,
  fillColor: '#ffff00'
});
var os = brd.createElement('riemannsum', [f, function() { return s.Value(); }, function() { return a.Value(); }, function() { return b.Value(); }]);
```

Fragen?



```
MathGraph.initBoard('box', {ax:  
  Element('slider', [[1,3], [5  
  Element('slider', [[1,2], [5  
  Element('slider', [[1,1], [5  
  }) { return Math.sin(x); }  
  createElement('functiongraph',  
  function() { return s.Value(); }, function()  
  function() {return a.Value();},  
  function() {return b.Value();}  
  fillColor: '#ffff00';
```



**Danke für Ihre
Aufmerksamkeit!**

<http://jsxgraph.org>

```
var brd = JSXGraph.initBoard('box', {ax: 1, ay: 1, bx: 5, by: 5});  
var s = brd.createElement('slider', [[1, 3], [5, 5]]);  
var a = brd.createElement('slider', [[1, 2], [5, 5]]);  
var b = brd.createElement('slider', [[1, 1], [5, 5]]);  
var f = function(x) { return Math.sin(x); };  
var plot = brd.createElement('functiongraph', {f: f});  
var riemannsum = brd.createElement('riemannsum', [f, s, a, b]);  
riemannsum.function() { return s.Value(); },  
riemannsum.function() { return a.Value(); },  
riemannsum.function() { return b.Value(); },  
riemannsum.fillColor: '#ffff00';
```