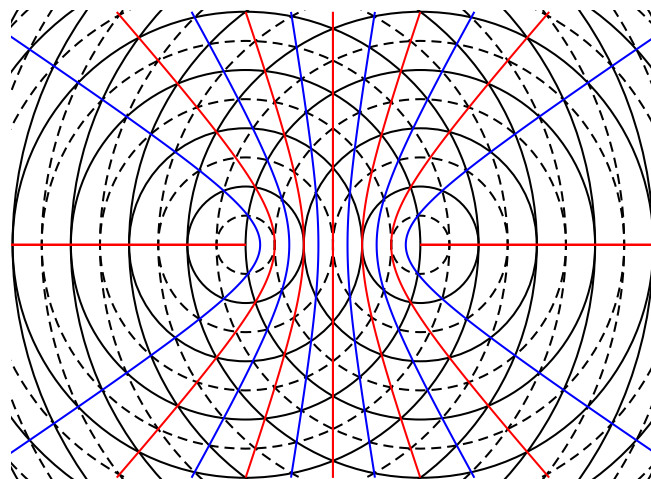


Skript zum Thema Interferenzhyperbeln

Xenia Rendtel

16. März 2010



Inhaltsverzeichnis

1	Motivation	2
2	Die Theorie	2
3	Programmeingabe	3
3.1	Beispiele	3
4	Programmcode	4

1 Motivation

Im Physik Oberstufenkurs wird im ersten Semester das Thema *Schwingungen und Wellen* bearbeitet. Dort treten dann natürlich auch Interferenzhyperbeln auf. Nachdem ich dies nun schon mehrfach unterrichtet habe, habe ich mich privat mit der zugehörigen Mathematik beschäftigt. Hierzu gibt es einen netten Artikel bei der Wikipedia.¹

Herausgekommen ist ein kleines Perl-Skript, mit dem ich Kreise und Hyperbeln mithilfe von Pstricks erzeuge.

2 Die Theorie

Hat man zwei Erregerzentren von Wellen, so ergeben sich für die Interferenzmuster Hyperbelfunktionen.

Eine Hyperbel ist definiert als die Menge aller Punkte der Zeichenebene, für die die Differenz der Abstände zu den Brennpunkten (erregerzentren) F_1 und F_2 konstant gleich $2a$ ist, wobei a die große Halbachse der Hyperbel ist.

$$\text{hyp} = \{P \mid |\overline{PF_1} - \overline{PF_2}| = 2a\} \quad (1)$$

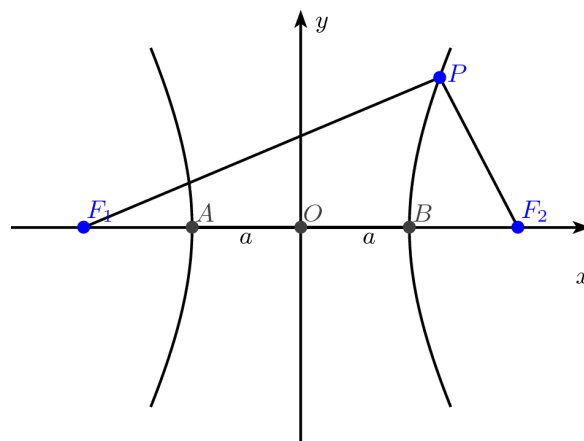


Abb. 1: Eine Hyperbel

Den halben Abstand der Brennpunkte voneinander bezeichnet man üblicherweise mit e . Die Gleichung der Hyperbel erhält eine besonders einfache Form, wenn sie in „1.Hauptlage“ liegt, das heißt, dass die beiden Brennpunkte auf der x -Achse symmetrisch zum Ursprung liegen. Also in den Punkten $(-e|0)$ und $(e|0)$.

Es gilt dann für die Hyperbel

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1 \quad (2)$$

wobei gilt:

$$a^2 + b^2 = e^2. \quad (3)$$

Hat man zwei Erreger mit der gleichen Wellenlänge, so liegen die Interferenzmaxima genau an den Stellen mit einem Gangunterschied von $\Delta s = n \cdot \lambda$, wobei λ die Wellenlänge ist und $n \in \mathbb{N}$.

¹Siehe (VERSCHIEDENE, 2010a) und (VERSCHIEDENE, 2010b)

D. h man sucht einen Punkt P auf der Interferenzhyperbel, der $d + \Delta s$ vom linken Erreger und d vom rechten Erreger entfernt ist, für die rechte Verstärkung.

Wobei oBdA $d = e$ ist. Es gilt dann:

$$\begin{aligned} \text{(I)} \quad (x + e)^2 + y^2 &= (d + s)^2 \\ \text{(II)} \quad (x - e)^2 + y^2 &= d^2 \end{aligned}$$

Daraus ergibt sich: $x = \frac{2ds+s^2}{4e}$ und $y = \sqrt{d^2 - (x - e)^2}$ Hieraus bestimmt man für die gesuchte Hyperbel a :

$$a = \frac{\sqrt{(x - e)^2 + y^2} - \sqrt{(x + e)^2 + y^2}}{2} \quad (4)$$

Setzt man dies nun in Gleichung (2) ein und löst dies nach y auf, so hat man die Funktion für die Hyperbel.

$$y = \pm \frac{\sqrt{(b^2 + e^2)(e^2 - x^2 + b^2)}b}{b^2 + e^2} \quad (5)$$

Mit einer geeigneten Parametrisierung erhält man:

$$t \mapsto (a \cosh(t), b \sinh(t)) \quad (6)$$

für $0 < t < 2\pi$.

3 Programmeingabe

Wie erhält man nun die gewünschten Ergebnisse? Hierzu übergibt man an mein Skript eine Datei mit dem folgenden Eingabeformat:

```
abstand 3
wellenlaenge 1,2
gegen
boegen
hyperbel
anzahlkreise 1
```

Zur Erklärung: `abstand n` gibt an, wie weit die Erregerzentren auf der x - Achse entfernt sind. `wellenlaenge l1, l2` gibt die Wellenlängen der zwei Erreger an. Setzt man `gegen`, so wird startet der zweite Erreger gegenphasig zum ersten Erreger. Wird `boegen` gesetzt, so werden weitere Kreisbögen gezeichnet. `anzahlkreise n` gibt an, wie viele Kreise gezeichnet werden. Ist dies nicht gesetzt, so werden 4 Kreise gemalt.

3.1 Beispiele

Als Beispiel hier ein paar Eingaben für mein Programm und deren Ausgaben:

1. Die Eingabe:

```
abstand 3
wellenlaenge 1,1
boegen
hyperbel
```

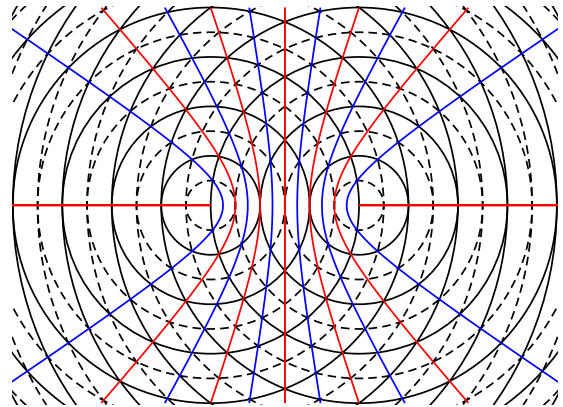


Abb. 2: Ein erstes Bild

2. Die Eingabe:

```
abstand 3
wellenlaenge 1,2
gegen
boegen
hyperbel
```

Das Bild

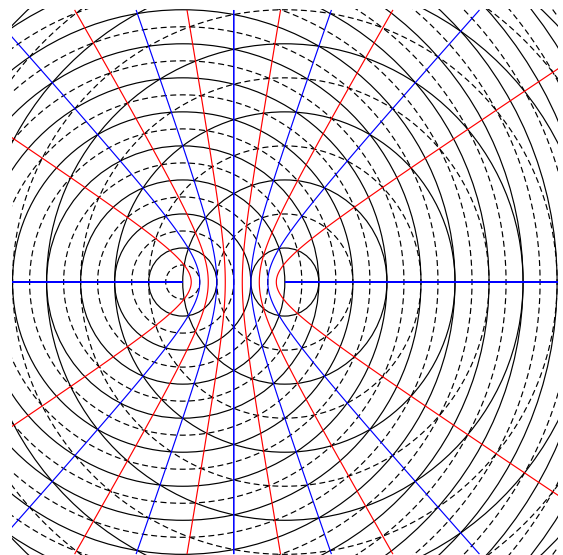


Abb. 3: Ein zweites Bild

4 Programmcode

Und hier nun der vollständige Programmcode in Perl:

```
1 #!/usr/bin/perl -w
2 #####
3 ## Programm zur Erstellung von Interferenzmustern mit Pstricks
4 ## Letzte Aenderung: 14.01.2010
5 ## Autor: Xenia Rendtel
6 #####
7 use POSIX qw /floor ceil/;
8 use Math::Trig;
9 use strict ;
10 use lib '/home/xenia/Schule/Programme/Interferenz';
11 use hilfsmfunktionen;
12 my (
13 $e,      $s,      $d,      $l1,  $l2,      $weiter,
```

```

14 $kreise, $diff, $i, $ymax, $ymin, $xmin,
15 $xmax, @ausgabedatei, @ausgabedateizeilen, $kopf, $kommandodatei, $texdatei,
16 $leer, $eingabedateizeilen, @eingabedatei, $kommandodateinummer, $datei,
17 $zeile, $ausrichtung, $bezeichnung, $func, $variable, $j,
18 $schritt,
19 $spalten, $nachkommastellen, $spaltenausrichtung, $liste, $senderreicht,
20 $hilfszeile,
21 $anzahlzeilen
22 );
23 my $boegen = 0;
24 my $hyperbel = 0;
25 $xmin = $xmax = $ymin = $ymax = 0;
26 $kopf = 1;
27 $diff = 0;
28 ### Das Array wird eingelesen
29 @ausgabedateizeilen = ();
30
31 for ( $i = 0 ; $i <= 150 ; $i++ ) { $ausgabedateizeilen[$i] = 0; }
32
33 # In ein array wird geschrieben
34 sub schreibeinarray {
35     my ( $layer, $zeile ) = @_;
36     $ausgabedatei[$layer][ $ausgabedateizeilen[$layer] ] = $zeile ;
37     $ausgabedateizeilen[$layer]++;
38 }
39
40 # Der Kopf der Latex-Datei wird geschrieben
41 sub kopfdatei {
42     if ( $kopf == 1 ) {
43         kommentar( 1, "Interferenzhyperbeln" );
44         kommentar( 1, "Autor: X. Rendtel" );
45         kommentar( 1, "Letzte Aenderung: 2009" );
46         latex( 2, "\\documentclass[10pt, a4paper]{article}" );
47         usepackage( "utf8", "inputenc" );
48         usepackage( "",
49             "pstricks, pst-pdf, pst-node, xcolor, "
50             . "pst-circ, pst-func, pst-math, pst-eucl, "
51             . "pstricks-add, multido" );
52         latex( 3, "\\pagestyle{empty} \\n\\begin{document}" );
53     }
54     beginpicture(4);
55 }
56
57 sub fussdatei {
58     endpicture(11);
59     if ( $kopf == 1 ) { latex( 11, "\\end{document}" ); }
60 }
61 #### Pictureumgebung
62 sub beginpicture {
63     my ( $layer ) = @_;
64     schreibeinarray(
65         $layer,
66         sprintf(
67             "\\begin{pspicture}(%g,%g)(%g,%g)\\n",
68             getminmax( "x", "min" ),
69             getminmax( "y", "min" ),
70             getminmax( "x", "max" ),
71             getminmax( "y", "max" )
72         )
73     );
74 }

```

```

75
76 sub endpicture {
77   my ($layer) = @_ ;
78   latex( $layer, "\\end{pspicture}");
79 }
80
81 # Die Eingabedatei wird gelesen
82 sub leseeingabedatei {
83   my $dateiname = shift(@_);
84   my $zeile;
85   open( my $EINGABEDATEI, $dateiname )
86     || die $dateiname . ": $!";
87   while ( $zeile = <$EINGABEDATEI> ) {
88
89     # Zeilenenden beseitigen, Kommentare und Leerzeilen ignorieren
90     $zeile =~ s/[\\r\\n]//g;
91     $zeile =~ s/ +/ /g;
92     $zeile =~ s/ +$/ /g;
93     $zeile =~ s/^ +//g;
94     next if ( $zeile =~ /^\\#/ );
95     next if ( $zeile =~ /^$/ );
96     if ( $zeile =~ /^lesedatei (.*)$/i ) {
97       leseeingabedatei($1);
98     }
99     else {
100      $eingabedatei[$eingabedateizeilen] = $zeile ;
101      $eingabedateizeilen++;
102    }
103  }
104  close($EINGABEDATEI);
105 }
106 ### Minimum und Maximum setzen
107 sub setminmax {
108   my ( $min, $xodery, $minmax ) = @_ ;
109   if ( $xodery eq "x" ) {
110     if ( $minmax eq "min" ) { $xmin = $min; }
111     else { $xmax = $min; }
112   }
113   if ( $xodery eq "y" ) {
114     if ( $minmax eq "min" ) { $ymin = $min; }
115     else { $ymax = $min; }
116   }
117 }
118
119 sub getminmax {
120   my ( $xodery, $minmax ) = @_ ;
121   if ( $xodery eq "x" ) {
122     if ( $minmax eq "min" ) { return $xmin; }
123     else { return $xmax; }
124   }
125   if ( $xodery eq "y" ) {
126     if ( $minmax eq "min" ) { return $ymin; }
127     else { return $ymax; }
128   }
129 }
130 ## Minimum und Maximum bestimmen
131 sub minmax {
132   my ( $xminlokal, $xmaxlokal, $yminlokal, $ymaxlokal ) = @_ ;
133   if ( $xminlokal < getminmax( "x", "min" ) ) {
134     setminmax( $xminlokal, "x", "min" );
135   }

```

```

136  if ( $xmaxlokal > getminmax("x", "max") ) {
137      setminmax( $xmaxlokal, "x", "max" );
138  }
139  if ( $yminlokal < getminmax("y", "min") ) {
140      setminmax( $yminlokal, "y", "min" );
141  }
142  if ( $ymaxlokal > getminmax("y", "max") ) {
143      setminmax( $ymaxlokal, "y", "max" );
144  }
145  }
146
147  sub degtorad {
148      my ($wert) = @_;
149      return $wert * pi() / 180;
150  }
151
152  sub radtodeg {
153      my ($wert) = @_;
154      return $wert * 180 / pi();
155  }
156  my $abstand = 0;
157
158  sub setabstand {
159      my ($wert) = @_;
160      $abstand = $wert;
161  }
162
163  sub getabstand { return $abstand; }
164
165  my ( $wellenlaenge1, $wellenlaenge2 );
166  $wellenlaenge1 = $wellenlaenge2 = 1;
167
168  sub setwelleeins {
169      my ($wert) = @_;
170      $wellenlaenge1 = $wert;
171  }
172
173  sub setwellezwei {
174      my ($wert) = @_;
175      $wellenlaenge2 = $wert;
176  }
177
178  sub getwelleeins { return $wellenlaenge1; }
179
180  sub getwellezwei { return $wellenlaenge2; }
181  my $anzahl = 4;
182
183  sub setkreisanzahl {
184      my ($wert) = @_;
185      $anzahl = $wert;
186  }
187
188  sub getkreisanzahl { return $anzahl; }
189  my $liniendicke = 1;
190
191  sub setliniendicke {
192      my ($wert) = @_;
193      $liniendicke = $wert;
194  }
195
196  sub getliniendicke { return $liniendicke; }

```

```

197
198 sub seterreger {
199   my ( $abstand, $wellenlaenge1, $wellenlaenge2, $anzahl, $liniendicke ) = @_ ;
200   $e = $abstand / 2;
201   kommentar( 2, "Abstand Erreger: " . $abstand );
202   kommentar( 2, "Wellenlaenge Erreger 1: " . $wellenlaenge1 );
203   kommentar( 2, "Wellenlaenge Erreger 2: " . $wellenlaenge2 );
204   kommentar( 2, "Anzahl Kreise: " . $anzahl );
205   ausgabe(
206     "\\psset{linecolor=black, fillcolor =black!20, "
207     . " linestyle=solid, linewidth= "
208     . $liniendicke
209     . "pt, dotstyle = *, plotpoints = 1000, dotsize = 3pt,"
210     . " arrowsize = 3pt 2, arrowinset = 0.25, xunit = 1cm, "
211     . "yunit = 1cm, algebraic =true}
212   "
213   );
214 }
215 sub setboegen { my $wert = @_ ; $boegen = $wert; }
216 sub getboegen { return $boegen; }
217 ## Die Kreise werden gezeichnet
218 sub erzeugekreise {
219   my ( $x, $lambda, $start, $anzahlkreise ) = @_ ;
220   minmax(
221     $x - $lambda * ($anzahlkreise),
222     $x + $lambda * ($anzahlkreise),
223     -$lambda * ($anzahlkreise),
224     $lambda * ($anzahlkreise)
225   );
226   if ( getboegen() == 1 ) { $anzahlkreise = 5 * $anzahlkreise; }
227   else { $anzahlkreise = $anzahlkreise; }
228   if ( $start == 0 ) {
229     schreibeinarray( 10,
230       sprintf( "\\multido{\\nx=0+%f}{%i}{", $lambda, $anzahlkreise ) );
231     schreibeinarray( 10,
232       sprintf( "\\pscircle[linewidth=1pt](%f,0){\\nx}\\n", $x ) );
233     schreibeinarray(
234       10,
235       sprintf(
236         "\\multido{\\nx=%f+%f}{%i}{",
237         $lambda / 2,
238         $lambda, $anzahlkreise
239       )
240     );
241     schreibeinarray(
242       10,
243       sprintf( "\\pscircle[ linestyle=dashed, linewidth=1pt](%f,0){\\nx}\\n",
244         $x )
245     );
246   }
247   else {
248     schreibeinarray( 10,
249       sprintf( "\\multido{\\nx=0+%f}{%i}{", $lambda, $anzahlkreise ) );
250     schreibeinarray(
251       10,
252       sprintf( "\\pscircle[ linestyle=dashed, linewidth=1pt](%f,0){\\nx}\\n",
253         $x )
254     );
255     schreibeinarray(
256       10,
257       sprintf(

```



```

258     "\\multido{\\nx=%f+%f}{%i}{",
259     $lambda / 2,
260     $lambda, $anzahlkreise
261     )
262 );
263 schreibeinarray( 10,
264     sprintf( "\\pscircle[linewidth=1pt](%f,0){\\nx}\\n", $x ) );
265 }
266 }
267
268 sub hyperbelfunktion {
269     my ( $a, $e, $ymin, $ymax, $farbe ) = @_;
270     my $b = sqrt( $e * $e - $a * $a );
271     my $i = 0;
272     my $t = 0;
273     for ( $i = 0 ; $i <= 3 ; $i = $i + 1 / 1000 ) {
274         if ( $b * sinh($i) < $ymax ) { $t = $i; }
275         else { $t = $t; }
276     }
277     schreibeinarray(
278         10,
279         sprintf(
280             "\\parametricplot[linecolor=%s]{%g}{%g}{%g*COSH(t)|%g*SINH(t)}",
281             $farbe, -$t, $t, $a, $b
282         )
283     );
284     schreibeinarray(
285         10,
286         sprintf(
287             "\\parametricplot[linecolor=%s]{%g}{%g}{%g*COSH(t)|%g*SINH(t)}",
288             $farbe, -$t, $t, -$a, -$b
289         )
290     );
291 }
292
293 sub kreisboegen {
294     my ( $x1, $laenge, $kreise, $start ) = @_;
295     my ( $xabstandlinks, $xabstandrechts, $alpha, $beta, $gamma, $delta, $i,
296         $abstand, $eckelinks, $eckerechts, $berg, $tal, );
297     if ( $start == 0 ) { $berg = "solid"; $tal = "dashed"; }
298     else { $berg = "dashed"; $tal = "solid"; }
299 }
300
301 sub kreisboegen2 {
302     my ( $x1, $laenge, $kreise, $start ) = @_;
303     my ( $xabstandlinks, $xabstandrechts, $alpha, $beta, $gamma, $delta, $i,
304         $abstand, $eckelinks, $eckerechts, $berg, $tal, );
305     if ( $start == 0 ) { $berg = "solid"; $tal = "dashed"; }
306     else { $berg = "dashed"; $tal = "solid"; }
307     $xabstandlinks = abs( $x1 - $xmin );
308     $xabstandrechts = abs( $xmax - $x1 );
309     $eckelinks = sqrt( $ymax * $ymax + $xabstandlinks * $xabstandlinks );
310     $eckerechts = sqrt( $ymax * $ymax + $xabstandrechts * $xabstandrechts );
311
312     for ( $i = 0 ; $i <= 100 ; $i++ ) {
313         $abstand = ( $kreise * $laenge ) + $laenge * $i;
314         if ( $abstand <= $xabstandlinks ) {
315             $beta = nachkommastellen( radtodeg( acos( $ymax / $abstand ) ), 4 );
316             if ( $abstand <= $eckelinks ) {
317                 schreibeinarray(
318                     10,

```

```

319     printf(
320         "\\psarc[linestyle=%s](%g,0){%g}{%g}{%g}",
321         $berg, $x1, $abstand, 90 + $beta, 270 - $beta
322     )
323 );
324 }
325 }
326 else {
327     $alpha =
328     nachkommastellen( radtodeg( acos( $xabstandlinks / $abstand ) ), 4 );
329     $beta = nachkommastellen( radtodeg( acos( $ymax / $abstand ) ), 4 );
330     if ( $abstand <= $eckelinks ) {
331         schreibeinarray(
332             10,
333             printf(
334                 "\\psarc[linestyle=%s](%g,0){%g}{%g}{%g}",
335                 $berg, $x1, $abstand, 90 + $beta, 180 - $alpha
336             )
337         );
338         schreibeinarray(
339             10,
340             printf(
341                 "\\psarc[linestyle=%s](%g,0){%g}{%g}{%g}",
342                 $berg, $x1, $abstand,
343                 180 + $alpha,
344                 270 - $beta
345             )
346         );
347     }
348 }
349 if ( $abstand <= $xabstandrechts ) {
350     $delta = nachkommastellen( radtodeg( acos( $ymax / $abstand ) ), 4 );
351     if ( $abstand <= $eckerechts ) {
352         schreibeinarray(
353             10,
354             printf(
355                 "\\psarc[linestyle=%s](%g,0){%g}{%g}{%g}",
356                 $berg, $x1, $abstand,
357                 270 + $delta,
358                 90 - $delta
359             )
360         );
361     }
362 }
363 else {
364     $gamma =
365     nachkommastellen( radtodeg( acos( $xabstandrechts / $abstand ) ), 4 );
366     $delta = nachkommastellen( radtodeg( acos( $ymax / $abstand ) ), 4 );
367     if ( $abstand <= $eckerechts ) {
368         schreibeinarray(
369             10,
370             printf(
371                 "\\psarc[linestyle=%s](%g,0){%g}{%g}{%g}",
372                 $berg, $x1, $abstand, $gamma, 90 - $delta
373             )
374         );
375         schreibeinarray(
376             10,
377             printf(
378                 "\\psarc[linestyle=%s](%g,0){%g}{%g}{%g}",
379                 $berg, $x1, $abstand,

```

```

380         270 + $delta,
381         360 - $gamma
382     )
383 );
384 }
385 }
386 }
387 for ( $i = 0 ; $i <= 100 ; $i++ ) {
388     $abstand = ( $kreise * $laenge + $laenge / 2 ) + $laenge * $i;
389     if ( $abstand <= $xabstandlinks ) {
390         $beta = nachkommastellen( radtodeg( acos( $ymax / $abstand ) ), 4 );
391         if ( $abstand <= $eckelinks ) {
392             schreibeinarray(
393                 10,
394                 sprintf(
395                     "\\psarc[linestyle=%s](%g,0){%g}{%g}{%g}",
396                     $tal, $x1, $abstand, 90 + $beta, 270 - $beta
397                 )
398             );
399         }
400     }
401     else {
402         $alpha =
403             nachkommastellen( radtodeg( acos( $xabstandlinks / $abstand ) ), 4 );
404         $beta = nachkommastellen( radtodeg( acos( $ymax / $abstand ) ), 4 );
405         if ( $abstand <= $eckelinks ) {
406             schreibeinarray(
407                 10,
408                 sprintf(
409                     "\\psarc[linestyle=%s](%g,0){%g}{%g}{%g}",
410                     $tal, $x1, $abstand, 90 + $beta, 180 - $alpha
411                 )
412             );
413             schreibeinarray(
414                 10,
415                 sprintf(
416                     "\\psarc[linestyle=%s](%g,0){%g}{%g}{%g}",
417                     $tal, $x1, $abstand,
418                     180 + $alpha,
419                     270 - $beta
420                 )
421             );
422         }
423     }
424     if ( $abstand <= $xabstandrechts ) {
425         $delta = nachkommastellen( radtodeg( acos( $ymax / $abstand ) ), 4 );
426         if ( $abstand <= $eckerechts ) {
427             schreibeinarray(
428                 10,
429                 sprintf(
430                     "\\psarc[linestyle=%s](%g,0){%g}{%g}{%g}",
431                     $tal, $x1, $abstand,
432                     270 + $delta,
433                     90 - $delta
434                 )
435             );
436         }
437     }
438     else {
439         $gamma = radtodeg( acos( $xabstandrechts / $abstand ) );
440         $delta = radtodeg( acos( $ymax / $abstand ) );

```

```

441     if ( $abstand <= $eckerechts ) {
442         schreibeinarray(
443             10,
444             sprintf(
445                 "\\psarc[linestyle=%s](%g,0){%g}{%g}{%g}",
446                 $tal, $x1, $abstand, $gamma, 90 - $delta
447             )
448         );
449         schreibeinarray(
450             10,
451             sprintf(
452                 "\\psarc[linestyle=%s](%g,0){%g}{%g}{%g}",
453                 $tal, $x1, $abstand,
454                 270 + $delta,
455                 360 - $gamma
456             )
457         );
458     }
459 }
460 }
461 ausgabe("");
462 }
463
464 sub hyperbel {
465     my ( $e, $laenge1, $laenge2, $anzahl ) = @_ ;
466     my ( $a, $s, $d, $n, $x, $y, $xhilf, $yhilf, $xnext, $ynext, $genau );
467     $genau = 0;
468     $d = ceil( $e );
469     for ( $n = 0 ; $n <= $anzahl ; $n++ ) {
470         $s = $laenge1 * $n;
471         $xhilf = nachkommastellen( ( 2 * $d * $s + $s * $s ) / ( 4 * $e ), 4 );
472         if ( ( $d * $d - ( $xhilf - $e ) * ( $xhilf - $e ) ) >= 0 ) {
473             $yhilf =
474                 nachkommastellen( sqrt( $d * $d - ( $xhilf - $e ) * ( $xhilf - $e ) ),
475                 4 );
476             $a = (
477                 sqrt( ( $xhilf - $e ) * ( $xhilf - $e ) + $yhilf * $yhilf ) -
478                 sqrt( ( $xhilf + $e ) * ( $xhilf + $e ) + $yhilf * $yhilf ) ) / 2;
479             if ( $diff == 1 ) {
480                 hyperbelfunktion( $a, $e, -$ymax, $ymax, "blue" );
481             }
482             else {
483                 hyperbelfunktion( $a, $e, -$ymax, $ymax, "red" );
484             }
485         }
486     }
487     for ( $n = 1 ; $n <= $anzahl ; $n++ ) {
488         $s = ( 2 * $n - 1 ) * $laenge1 / 2;
489         $xhilf = ( 2 * $d * $s + $s * $s ) / ( 4 * $e );
490         if ( ( $d * $d - ( $xhilf - $e ) * ( $xhilf - $e ) ) >= 0 ) {
491             $yhilf = sqrt( $d * $d - ( $xhilf - $e ) * ( $xhilf - $e ) );
492             $a = (
493                 sqrt( ( $xhilf - $e ) * ( $xhilf - $e ) + $yhilf * $yhilf ) -
494                 sqrt( ( $xhilf + $e ) * ( $xhilf + $e ) + $yhilf * $yhilf ) ) / 2;
495             if ( $diff == 1 ) {
496                 hyperbelfunktion( $a, $e, -$ymax, $ymax, "red" );
497             }
498             else {
499                 hyperbelufnktion( $a, $e, -$ymax, $ymax, "blue" );
500             }
501         }

```

```

502 }
503 }
504 my $start1 = 0;
505 my $start2 = 0;
506 $kommandodateinummer = 0;
507 while ( exists $ARGV[$kommandodateinummer] ) {
508     $kommandodatei = $ARGV[$kommandodateinummer];
509     $texdatei      = $kommandodatei;
510     $texdatei =~ s/\.ptxt$/i;
511     $texdatei = $texdatei . " . adam ";
512     $kommandodateinummer++;
513     $eingabedateizeilen = 0;
514     leseingabedatei($kommandodatei);
515     for ( $i = 0 ; $i < $eingabedateizeilen ; $i++ ) {
516         $zeile = $eingabedatei[$i];
517         if ( $zeile =~ /^abstand( [-0-9\.]+)$/i ) {
518             setabstand($1);
519         }
520         elsif ( $zeile =~ /^anzahlkreise( [-0-9\.]+)$/i ) {
521             setkreisanzahl($1);
522         }
523         elsif ( $zeile =~ /^wellenlaenge( [-0-9\.]+),([-0-9\.]+)$/i ) {
524             setwelleeins($1);
525             setwellezwei($2);
526         }
527         elsif ( $zeile =~ /^gegen$/i ) { $start2 = 1; $diff = 1; }
528         elsif ( $zeile =~ /^boegen$/i ) { setboegen(1); }
529         elsif ( $zeile =~ /^hyperbel$/i ) { $hyperbel = 1; }
530
531         # Alle anderen Befehle sind Schrott
532         else { printf( "%s\n", $zeile ); }
533     }
534 }
535 seterreger(
536     getabstand(), getwelleeins(), getwellezwei(), getkreisanzahl(),
537     getliniendicke()
538 );
539 erzeugekreise( -$e, getwelleeins(), $start1, getkreisanzahl() );
540 erzeugekreise( $e, getwellezwei(), $start2, getkreisanzahl() );
541 if ( $hyperbel == 1 ) {
542     hyperbel( $e, getwelleeins(), getwellezwei(), 1000 );
543 }
544 $leer = 1;
545 my $layer;
546 for ( $layer = 0 ; $layer <= 100 ; $layer++ ) {
547
548     for ( $i = 0 ; $i < $ausgabedateizeilen[$layer] ; $i++ ) {
549         $leer = 0;
550     }
551 }
552 if ( $leer == 0 ) {
553     kopfdatei();
554     fussdatei();
555     for ( $layer = 0 ; $layer <= 11 ; $layer++ ) {
556         for ( $i = 0 ; $i < $ausgabedateizeilen[$layer] ; $i++ ) {
557             printf( "%s\n", $ausgabedatei[$layer][$i] );
558         }
559     }
560 }
561 else { printf( "%s\n", "Die Datei ist leer\n" ); }

```

Literatur

- [BRONŠTEJN und SEMENDJAEV, 1985] BRONŠTEJN, IL'JA und K. A. SEMENDJAEV (1985). Taschenbuch der Mathematik. Teubner [u.a.], Leipzig [u.a.], 22 Aufl.
- [GREHN und KRAUSE, 1998] GREHN, JOACHIM und J. KRAUSE (1998). Metzler Physik Sekundarstufe II - 3. Auflage: Metzler Physik (3. A.). Gesamtband.. Schroedel, 3. A., 3. Nachdruck 2000. Aufl.
- [TIPLER, 2000] TIPLER, PAUL (2000). Physik. Spektrum Akad. Verl., Heidelberg [u.a.], 3., korrigierter Nachdr. der 1. Aufl. 1994. Aufl.
- [VERSCHIEDENE, 2010a] VERSCHIEDENE (2010a). Ellipse. <http://de.wikipedia.org/wiki/Ellipse>.
- [VERSCHIEDENE, 2010b] VERSCHIEDENE (2010b). Hyperbel (Mathematik). [http://de.wikipedia.org/wiki/Hyperbel_\(Mathematik\)](http://de.wikipedia.org/wiki/Hyperbel_(Mathematik)).