

Menggambar Grafik 2D dengan EMT

Basic Plots

Ada fungsi dasar dari plot. Ada koordinat layar, yang selalu berkisar dari 0 hingga 1024 pada setiap sumbu, tidak peduli apakah layarnya berbentuk persegi atau tidak. Dan ada koordinat plot, yang dapat diatur dengan `setplot()`. Pemetaan antara koordinat tergantung pada jendela plot saat ini. Misalnya, `shrinkwindow()` default menyisakan ruang untuk label sumbu dan judul plot.

Dalam contoh ini, kita dapat menggambar beberapa garis acak dalam berbagai warna.

```
>clg; // clear screen
>setplot(0,1,0,1); // set plot coordinates
>hold on; // start overwrite mode
>n=100; X=random(n,2); Y=random(n,2); // get random points
>colors=rgb(random(n),random(n),random(n)); // get random colors
>loop 1 to n; color(colors[#]); plot(X[#],Y[#]); end; // plot
>hold off; // end overwrite mode
>insimg; // insert to notebook
>reset();
```

Penting untuk menahan (hold on) grafik, karena perintah `plot()` akan membersihkan jendela plot. Untuk menghapus semua yang telah kita lakukan, kita menggunakan `reset()`.

Untuk menampilkan gambar hasil plot di layar notebook, perintah `plot2d()` dapat diakhiri dengan titik dua (`:`).

ara lain adalah perintah `plot2d()` diakhiri dengan titik koma (`;`), kemudian menggunakan perintah `insimg()` untuk menampilkan gambar hasil plot.

Untuk contoh lainnya, kita menggambar sebuah plot sebagai inset di dalam plot lain. Ini dilakukan dengan mendefinisikan jendela plot yang lebih kecil.

Perlu diperhatikan bahwa jendela ini tidak menyediakan ruang untuk label sumbu di luar jendela plot. Kita harus menambahkan margin jika diperlukan. Perhatikan juga bahwa kita menyimpan dan mengembalikan jendela penuh, serta menahan plot saat ini saat kita membuat plot inset

```
>clg;
>plot2d("x^3-x");
>xw=200; yw=100; ww=300; hw=300;
>ow=window();
>window(xw,yw,xw+ww,yw+hw);
>hold on;
>barclear(xw-50,yw-10,ww+60,ww+60);
>plot2d("x^4-x",grid=6);
>hold off;
>window(ow);
```

Plot default menggunakan jendela plot berbentuk persegi. Anda dapat mengubahnya dengan fungsi `aspect()`. Jangan lupa untuk mengatur ulang aspek tersebut setelahnya. Anda juga dapat mengubah pengaturan default ini di menu dengan "Set Aspect" ke rasio aspek tertentu atau sesuai ukuran jendela grafik saat ini.

Namun, Anda juga bisa mengubahnya hanya untuk satu plot. Untuk ini, ukuran area plot saat ini diubah, dan jendela diatur agar label memiliki cukup ruang.

```
>clg;  
>aspect(2); // rasio panjang dan lebar 2:1  
>plot2d(["sin(x)","cos(x)"],0,2pi):  
>aspect();  
>reset;
```

Fungsi `reset()` mengembalikan pengaturan plot default, termasuk rasio aspek.

Membuat 2D Plots di Euler

EMT Math Toolbox memiliki plot dalam 2D, baik untuk data maupun fungsi. EMT menggunakan fungsi `plot2d`. Fungsi ini dapat memplot fungsi dan data.

Dimungkinkan untuk memplot di Maxima menggunakan Gnuplot atau di Python menggunakan Math Plot Lib.

Euler dapat memplot grafik 2D dari:

- ekspresi,
- fungsi, variabel, atau kurva parameter,
- vektor nilai x-y,
- kumpulan titik-titik di bidang,
- kurva implisit dengan level atau wilayah level,
- Fungsi kompleks.

Gaya plot mencakup berbagai gaya untuk garis dan titik, plot batang, dan plot berbayang.

Grafik Fungsi Satu Variabel dalam Bentuk Ekspresi langsung

```
>plot2d("x^2");  
>plot2d("x^2")  
>clg  
>aspect(1.5); plot2d("x^3-x")  
>reset;
```

Grafik Fungsi Satu Variabel Tidak Langsung

Grafik Fungsi Satu Variabel yang Rumusnya Disimpan dalam variabel Ekspresi jika fungsinya akan digunakan berkali-kali atau untuk tujuan yang berbeda-beda.

```
>fx &= x^2-2*x-3;  
>plot2d(fx):
```

Contoh Grafik Lain

```
>a:=5.6;  
>plot2d("exp(-a*x^2)/a"); insimg(30); // menampilkan gambar hasil plot setinggi 25
```

Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

Rentang plot diatur dengan parameter yang ditetapkan berikut:

- a, b: rentang x (default: -2, 2)
- c, d: rentang y (default: skala dengan nilai)
- r: alternatifnya, radius di sekitar pusat plot
- cx, cy: koordinat pusat plot (default: 0, 0)

```
>plot2d("x^3-x",-1,2):  
>plot2d("sin(x)",-2*pi,2*pi): // plot sin(x) pada interval [-2pi, 2pi]  
>plot2d("cos(x)","sin(3*x)",xmin=0,xmax=2pi):
```

$x_{\min}=0$ dan $x_{\max}=2\pi$ menentukan rentang variabel x dari 0 hingga 2π (sekitar 6.28), yang merupakan satu periode penuh untuk fungsi trigonometri.

Alternatif dari titik dua adalah perintah `insimg(lines)`,

yang menyisipkan plot dengan menempati jumlah baris teks yang ditentukan.

Di opsi, plot dapat diatur untuk muncul:

- di jendela terpisah yang dapat diubah ukurannya,
- di jendela notebook.

Lebih banyak gaya dapat dicapai dengan perintah plot khusus.

Jika plot tersembunyi, tekan tombol tabulator untuk melihatnya.

Untuk membagi jendela menjadi beberapa plot, gunakan perintah `figure()`. Dalam contoh ini, kita memplot x^1 hingga x^4 ke dalam 4 bagian jendela. `figure(0)` mengatur ulang jendela default.

```
>reset;  
>figure(2,2); ...  
>for n=1 to 4; figure(n); plot2d("x^"+n); end; ...  
>figure(0):
```

Dalam `plot2d()`, tersedia gaya alternatif dengan `grid=x`. Untuk gambaran umum, kami menampilkan berbagai gaya grid dalam satu gambar (lihat di bawah untuk perintah `figure()`). Gaya `grid=0` tidak disertakan. Gaya ini tidak menampilkan grid maupun bingkai.

```
>figure(3,3); ...  
>for k=1:9; figure(k); plot2d("x^3-x",-2,1,grid=k); end; ...  
>figure(0):
```

Jika argumen untuk `plot2d()` adalah sebuah ekspresi yang diikuti oleh empat angka, angka-angka tersebut merupakan rentang x dan y untuk plot.

Sebagai alternatif, a, b, c, d dapat ditentukan sebagai parameter yang ditetapkan, misalnya `a=...` dan seterusnya.

Dalam contoh berikut, kami mengubah gaya grid, menambahkan label, dan menggunakan label vertikal untuk sumbu y.

```
>aspect(1.5); plot2d("sin(x)",0,2pi,-1.2,1.2,grid=3,xl="x",yl="sin(x)");  
>plot2d("sin(x)+cos(2*x)",0,4pi):
```

Gambar-gambar yang dihasilkan dengan menyisipkan plot ke dalam jendela teks disimpan di direktori yang sama dengan notebook, secara default di subdirektori bernama "images". Gambar-gambar ini juga digunakan oleh ekspor HTML. Anda dapat dengan mudah memilih gambar apa pun dan menyalinnya ke clipboard dengan Ctrl-C.

Tentu saja, Anda juga dapat mengekspor grafik saat ini dengan fungsi-fungsi di menu File. Fungsi atau ekspresi dalam `plot2d` dievaluasi secara adaptif. Untuk kecepatan yang lebih tinggi, matikan plot adaptif dengan `<adaptive` dan tentukan jumlah subinterval dengan `n=...`. Ini hanya perlu dilakukan dalam kasus-kasus yang jarang.

```
>plot2d("sign(x)*exp(-x^2)",-1,1,<adaptive,n=10000):  
>plot2d("x^x",r=1.2,cx=1,cy=1):
```

Perlu dicatat bahwa x^x tidak terdefinisi untuk $x = 0$. Fungsi `plot2d` menangkap kesalahan ini, dan mulai memplot begitu fungsi tersebut terdefinisi. Ini berlaku untuk semua fungsi yang mengembalikan NAN di luar rentang definisinya.

```
>plot2d("log(x)",-0.1,2):
```

Parameter `square=true` (atau `>square`) secara otomatis memilih rentang y sehingga hasilnya adalah jendela plot berbentuk persegi. Perlu dicatat bahwa secara default, Euler menggunakan ruang berbentuk persegi di dalam jendela plot.

```
>plot2d("x^3-x",>square):  
>plot2d(''integrate("sin(x)*exp(-x^2)",0,x)'',0,2): // plot integral
```

Jika Anda memerlukan lebih banyak ruang untuk label sumbu y, panggil `shrinkwindow()` dengan parameter `smaller`, atau tetapkan nilai positif untuk "smaller" di `plot2d()`.

```
>plot2d("gamma(x)",1,10,yl="y-values",smaller=6,<vertical):
```

Ekspresi simbolik juga dapat digunakan, karena disimpan sebagai ekspresi string sederhana.

```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```

Membuat array x dari 0 hingga 2π dengan 1000 titik, kemudian memplot fungsi parametrik dengan $\sin(5x)$ di sumbu x dan $\cos(7x)$ di sumbu y .

```
>a:=5.6; expr &= exp(-a*x^2)/a; // define expression
```

Menetapkan nilai a menjadi 5.6 dan mendefinisikan ekspresi expr sebagai $\exp(-a*x^2)/a$, yang merupakan distribusi Gaussian.

```
>plot2d(expr,-2,2): // plot from -2 to 2
```

Memplot ekspresi expr dalam rentang x dari -2 hingga 2.

```
>plot2d(expr,r=1,thickness=2): // plot in a square around (0,0)
```

Memplot ekspresi expr di sekitar pusat plot $(0,0)$ dengan radius 1 dan tebal garis 2.

```
>plot2d(&diff(expr,x),>add,style="--",color=red): // add another plot
```

Memplot turunan pertama dari expr terhadap x dan menambahkannya ke plot yang ada dengan gaya garis putus-putus berwarna merah.

```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot in rectangle
```

Memplot turunan kedua dari expr terhadap x dalam sebuah persegi panjang dengan rentang x dari -2 hingga 2 dan y dari -2 hingga 1.

```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // keep plot square
```

Memplot turunan pertama dari expr terhadap x dalam rentang x dari -2 hingga 2 dan memastikan plot tetap berbentuk persegi.

```
>plot2d("x^2",0,1,steps=1,color=red,n=10):
```

Memplot fungsi x^2 dari 0 hingga 1 dengan satu langkah besar dan 10 titik, menggunakan warna merah.

```
>plot2d("x^2",>add,steps=2,color=blue,n=10):
```

Menambahkan plot baru untuk x^2 dengan dua langkah besar dan 10 titik, menggunakan warna biru.

Pra-Subtopik - Angka Kompleks

Ini adalah pengantar ke dalam aritmatika bilangan kompleks Euler. Sebagian besar fungsi numerik juga didefinisikan untuk bilangan kompleks.

```
>z=1+1i; exp(z^2);
```

Fungsi log membutuhkan perhatian, karena hanya didefinisikan untuk bidang tanpa sumbu negatif. Ini tidak berkesinambungan di seluruh sumbu negatif.

```
>log(-1+0.1i), log(-1-0.1i);
```

```
0.00497517+3.04192i
```

Kita dapat memplot bagian nyata dari kurva kompleks seperti biasa.

```
>plot2d("re(exp(3*I*x))",a=0,b=2*pi);
```

Fungsi `arg` menghitung argumen bilangan kompleks. Ini adalah, dari tentu saja, terputus-putus.

```
>plot2d("arg(exp(3*1i*x))",a=0,b=2*pi);
```

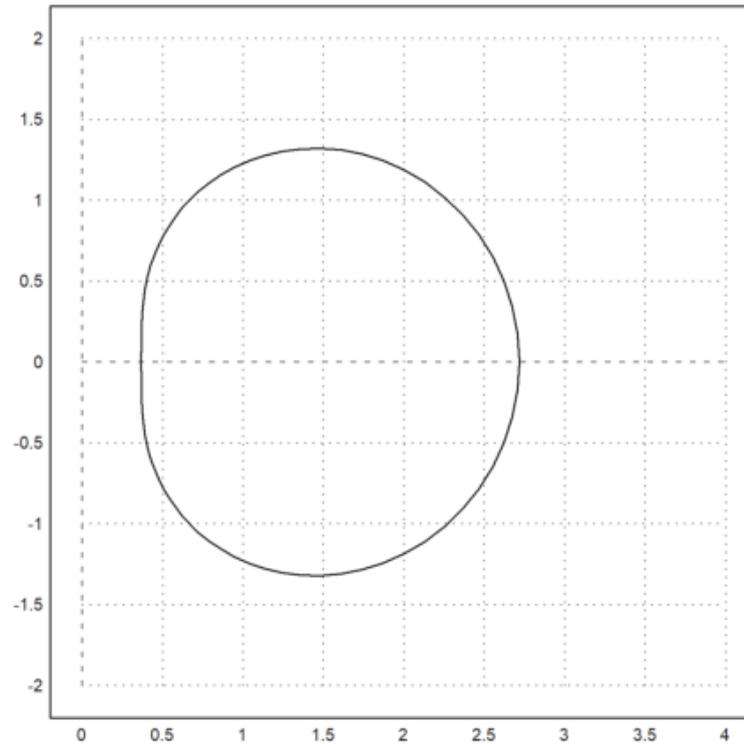
Subtopik - Fungsi Kompleks

Jadi, di EMT bisa lho membuat tabel fungsi untuk fungsi kompleks di sepanjang sebuah kurva. Dalam contoh ini, mari kita buat 500 titik yang tersebar di sekitar lingkaran satuan.

```
>z=exp(1i*linspace(0,2*pi,100));
```

Sekarang kita dapat dengan mudah memplot gambar lingkaran di bawah fungsi eksp.

```
>plot2d(exp(z),a=0,b=4,c=-2,d=2):
```



Kita juga dapat memplot gambar lengkap dari lingkaran satuan di bawah ini Pemetaan.

Pertama, kita perlu mendefinisikan vektor baris jari-jari dari 0 hingga 1 dan kolom vektor sudut dari 0 hingga 2π .

```
>r=linspace(0,1,190); phi=linspace(0,2*pi,800)';
```

Kita sekarang dapat menghasilkan grid yang menutupi lingkaran satuan,

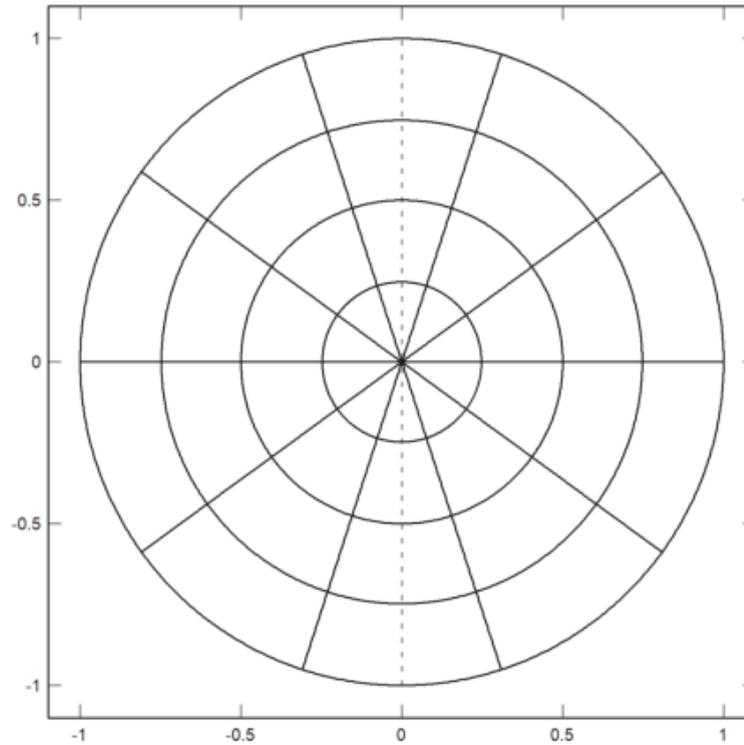
dan kita dapat memplot dengan mudah.

Fungsi `plot2d` memplot kisi. Secara default, semua titik grid adalah

Diplot. Tetapi kita juga dapat menentukan sejumlah titik grid dengan `cgrid=...` Untuk tampilan yang mulus, angka ini harus membagi jumlah poin secara merata.

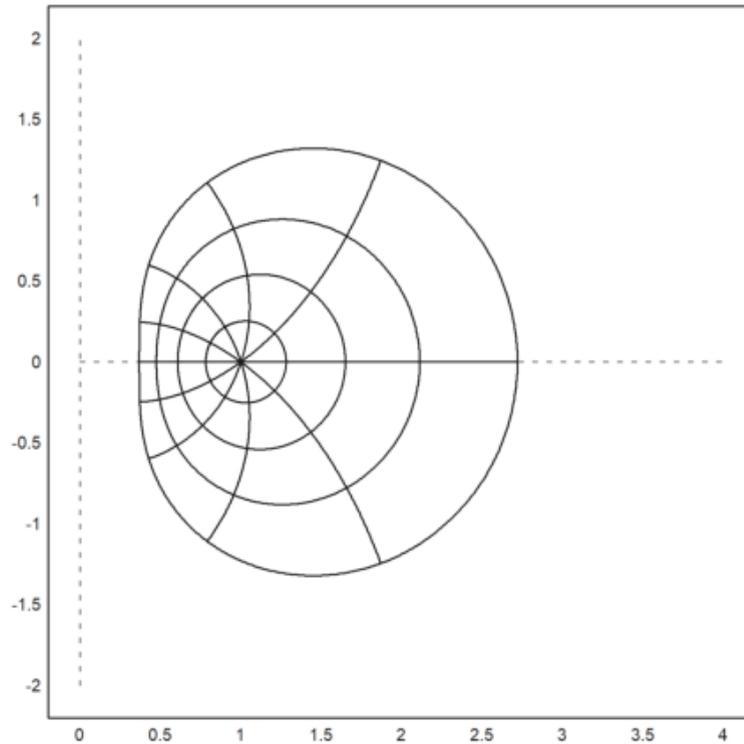
`grid=1` hanya mematikan grid koordinat.

```
>z=r*exp(phi*I); plot2d(z,a=-1,b=1,c=-1,d=1,grid=6,cgrid=[10,4]):
```



Sekarang kita memplot gambar grid ini di bawah fungsi exp.

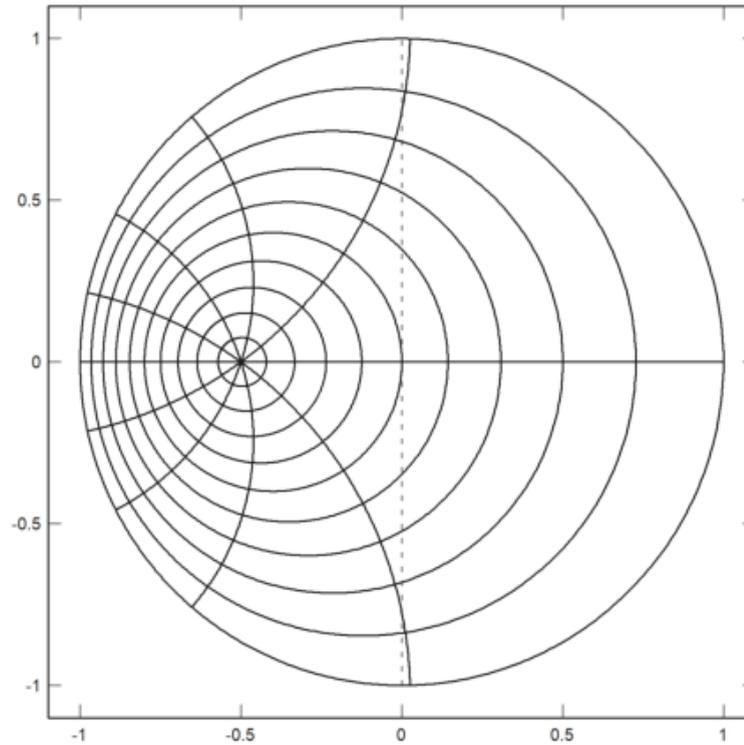
```
>plot2d(exp(z), a=0, b=4, c=-2, d=2, grid=1, cgrid=[10,4]):
```



Plot berikut menunjukkan pemetaan konformal dari lingkaran satuan ke Sendiri.

$$w = \frac{z - \frac{1}{2}}{1 - \frac{z}{2}}$$

```
>plot2d((z-0.5)/(1-0.5*z),a=-1,b=1,c=-1,d=1,grid=6,cgrid=10):
```



Berikut ini adalah pemetaan Joukowski yang memetakan bagian luar lingkaran satuan ke luar interval satuan $[-1,1]$ satu-ke-satu.

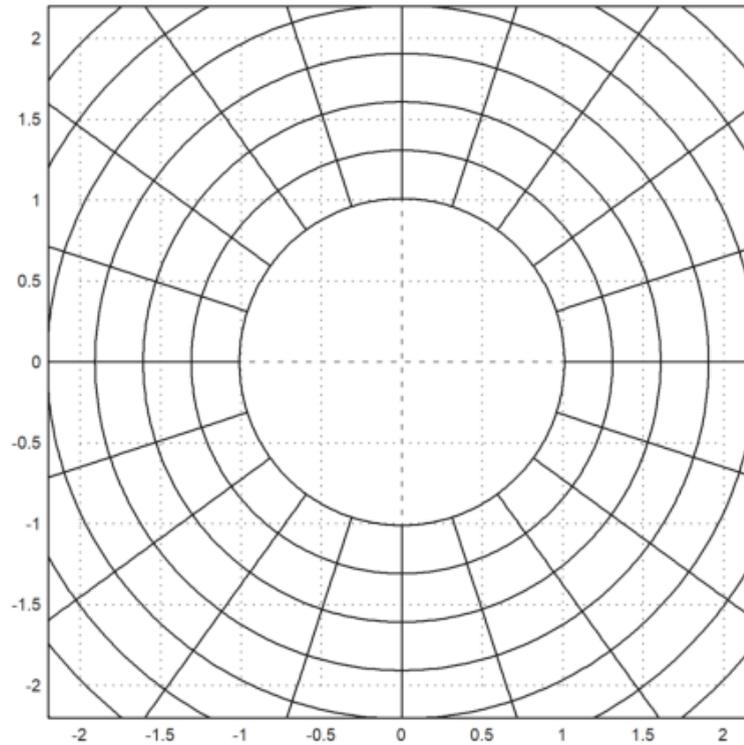
```
>function J(z) := (z+1/z)/2
```

Kami ingin menampilkan pemetaan di luar lingkaran satuan. Jadi kita perlu melakukannya menghasilkan grid di sana.

```
>z=linspace(1.01,4,100)*exp(1i*linspace(0,2*pi,400)');
```

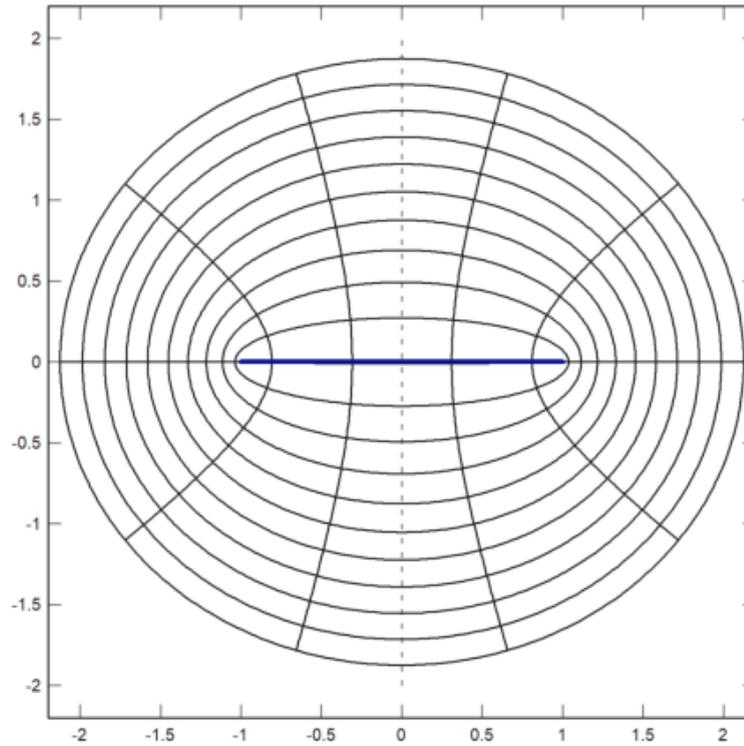
Pertama kita melihat gridnya.

```
>plot2d(z,a=-2,b=2,c=-2,d=2,grid=2,cgrid=[20,10]):
```



Kami memetakan grid dan memplot hasilnya, menambahkan interval $[-1,1]$ ke dalamnya biru pekat.

```
>plot2d(J(z),a=-2,b=2,c=-2,d=2,grid=6,cgrid=10); ...  
>plot2d([-1,1],0,color=blue,thickness=3,add=true):
```



Mari kita definisikan kebalikan dari pemetaan Joukowski. Kita harus menggunakan kasus di sini.

```
>function map invJ(z)...
```

```
w=sqrt(z^2-1);
$return z+(re(z)>=0)*w-(re(z)<0)*w;
$endfunction
```

Anda dapat menghitung kedua cabang secara simbolis.

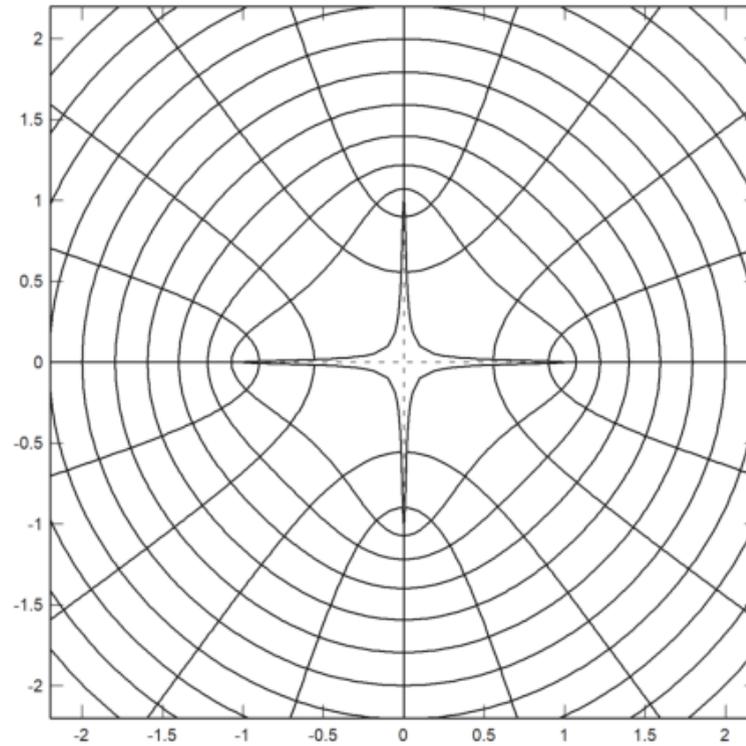
```
>&solve((z+1/z)/2=w,z)
```

```
(%o52) [z = w - sqrt(w^2 - 1), z = sqrt(w^2 - 1) + w]
```

Dengan menggunakan fungsi ini, kita dapat memetakan bagian luar lingkaran satuan bagian luar garis.

$$w = J(iJ^{-1}(\sqrt{2}J(z)))$$

```
>plot2d(J(I*invJ(J(z)*sqrt(2))),a=-2,b=2,c=-2,d=2,grid=6,cgrid=[20,10]):
```

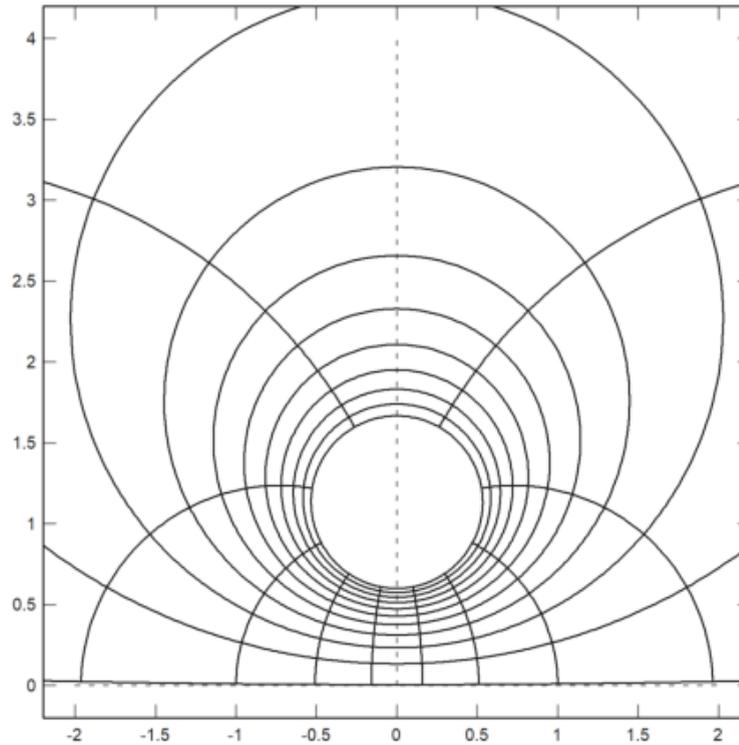


Kita dapat melihat gambar grid kita. Perhatikan bahwa tak terhingga dipetakan ke i.

```
>function H(z) &= (I*z+1)/(I+z)
```

```
(%o62)          I z + 1  
              -----  
              z + I
```

```
>plot2d(H(z), a=-2, b=2, c=0, d=4, grid=6, cgrid=10) :
```



Ini adalah fungsi kebalikannya, yang memetakan setengah bidang atas ke di luar lingkaran satuan.

```
>function invH(z) &= (I*z-1)/(I-z)
```

(%o68)

$$\frac{I z - 1}{I - z}$$

```
>&solve(w=H(z),z)
```

(%o74)

$$[z = \frac{1 - I w}{w - I}]$$

```
>invH(H(2)), invH(H(1i))
```

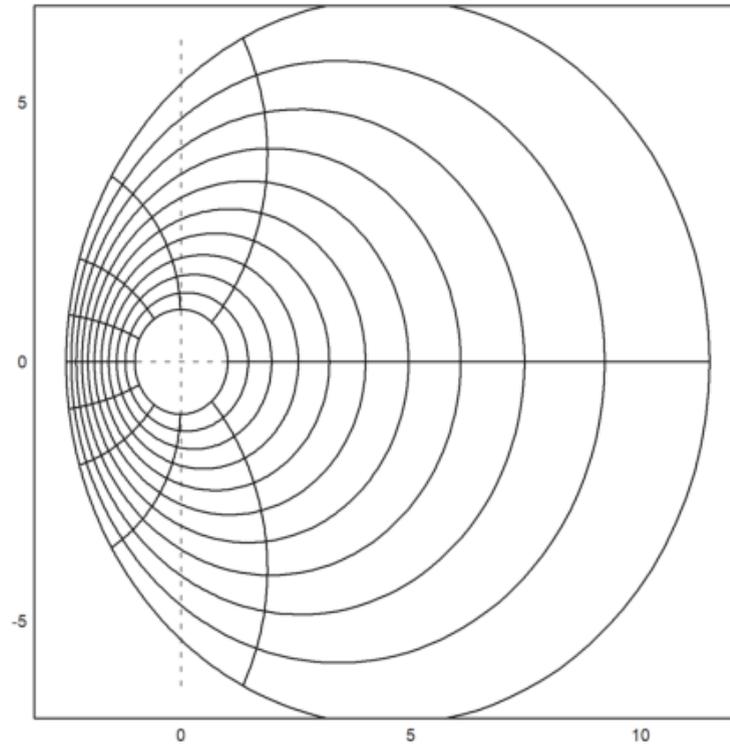
2+0i
0+1i

Fungsi selanjutnya memetakan bagian luar lingkaran satuan ke dirinya

sendiri, dan mengambil a hingga tak terhingga.

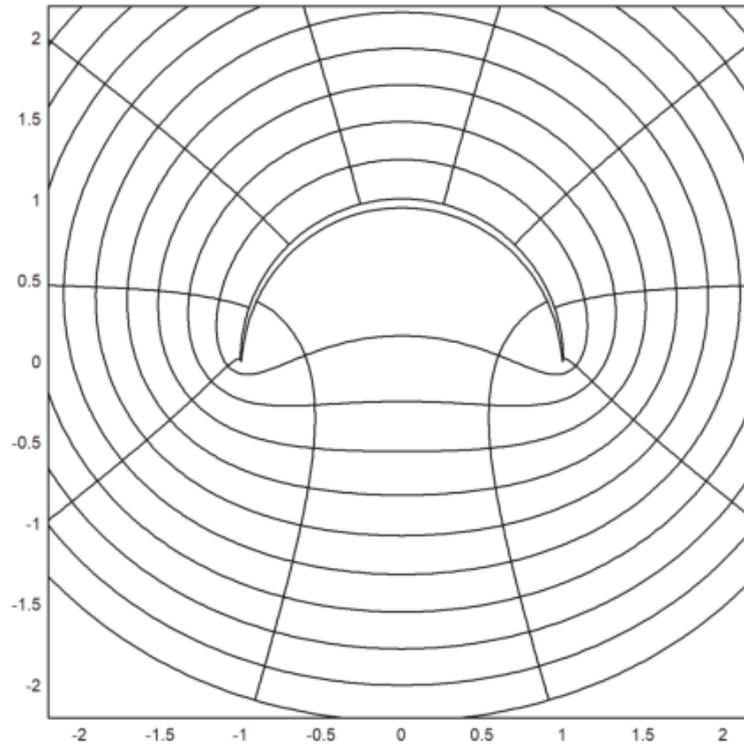
```
latex: w = L(z,a) = \frac{a}{\bar{a}} \frac{\bar{a}z-1}{a-z}
```

```
>function L(z,a) := (a/conj(a))*(conj(a)*z-1)/(a-z)  
>plot2d(L(z,6),grid=1,cgrid=10):
```



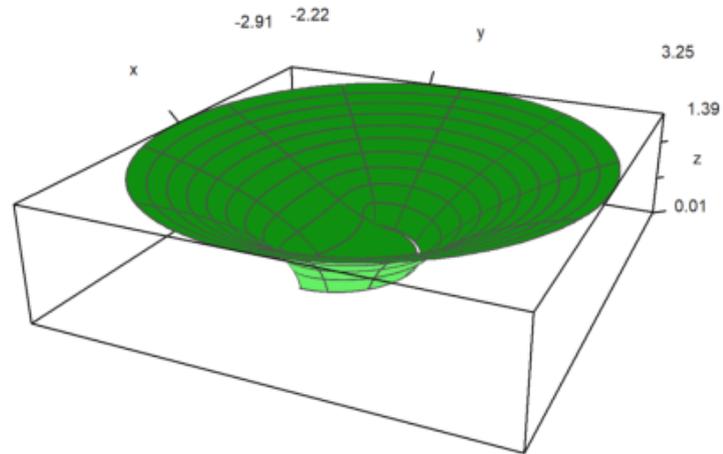
Sekarang kita memetakan bagian luar lingkaran satuan ke bagian luar busur dengan sudut π .

```
>plot2d(invH(J(L(z,-invJ(1i))))),a=-2,b=2,c=-2,d=2,grid=4,cgrid=10):
```



Mari kita plot fungsi Hijau dari busur ini dalam 3d.

```
>w=invH(J(L(z,-invJ(1i))));  
>plot3d(re(w),im(w),log(abs(z)),scale=1.2,grid=10):
```



Latihan Soal

Setelah teman-teman mempelajari dari materi yang sudah diberikan, sekarang waktunya mencoba latihan soal.

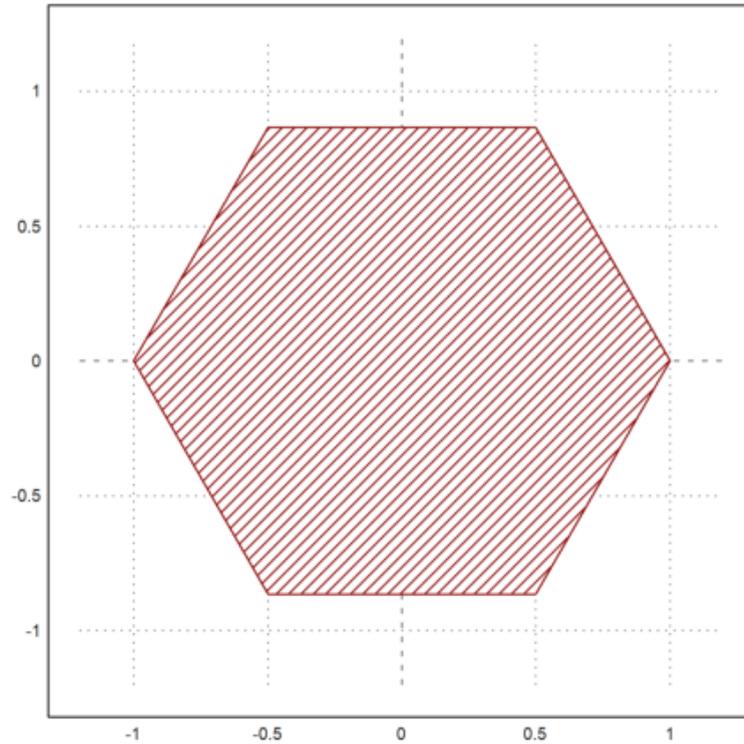
1. Buatlah segi enam menggunakan kurva tertutup dengan 6 poin dari

$$x = \cos(t)$$

$$y = \sin(t)$$

Kemudian "fill" plot segi enam tersebut.

```
>t=linspace(0,2pi,6); ...  
>plot2d(cos(t),sin(t),>filled,style="/",fillcolor=red,r=1.2):
```



Soal 2. Plotlah nilai polynomial dibawah ini dengan nilai
 $n=10; i=(0:n)$

$$B_i(x) = \binom{n}{i} x^i (1-x)^{n-i}$$

```
>x=0:0.01:1; n=10; i=(0:n)'; y=bin(n,i)*x^i*(1-x)^(n-i); ...  
>plot2d(x,y):
```

