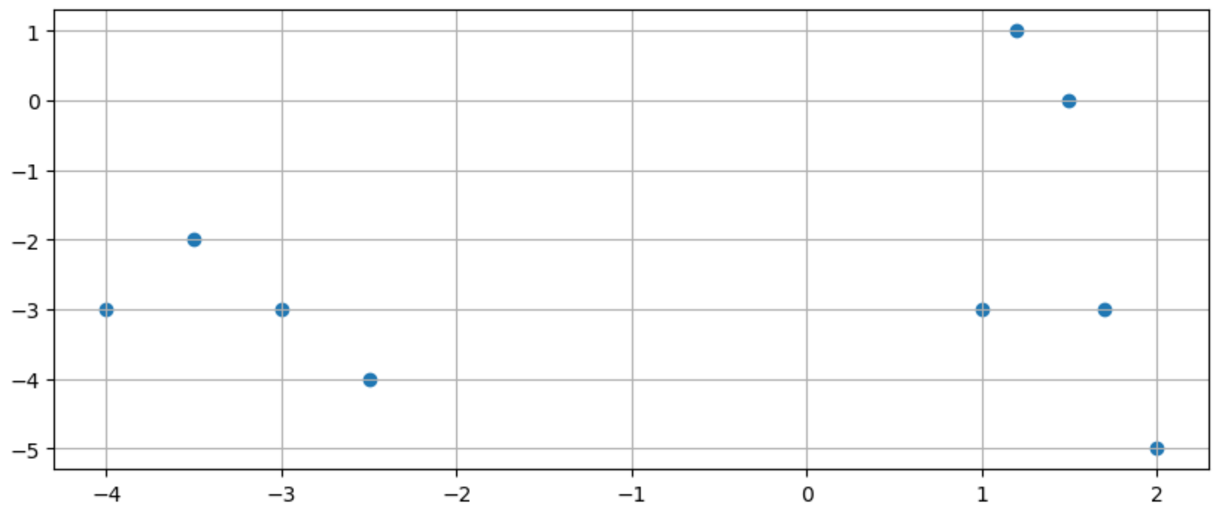
In [4]:
```
using PyPlot
using LinearAlgebra
```

Question 1

In [5]:
```
x_k = [-4;-3.5;-3;-2.5;1;1.2;1.5;1.7;2];
```

In [6]:
```
y_k = [-3;-2;-3;-4;-3;1;0;-3;-5];
```

In [7]:
```
figure(figsize=(10,4));
scatter(x_k,y_k);
grid();
```



Quadratic Form

$$f(x) = ax^2 + bx + c$$

Error

$$\varphi(\alpha) = ||A\alpha - B||^2$$

In [8]:
```
A = [x_k.^2 x_k ones(9)]
```

Out[8]:
```
9×3 Matrix{Float64}:
 16.0    -4.0  1.0
 12.25   -3.5  1.0
  9.0    -3.0  1.0
  6.25   -2.5  1.0
  1.0     1.0  1.0
  1.44    1.2  1.0
  2.25    1.5  1.0
  2.89    1.7  1.0
  4.0     2.0  1.0
```

In [9]:
```
alpha = A\y_k
```

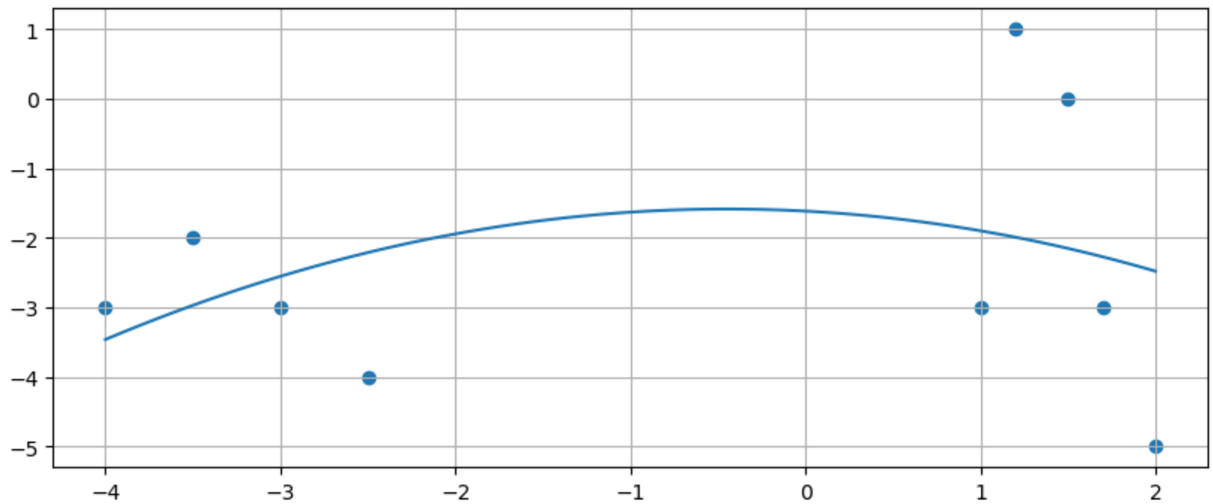
Out[9]:
```
3-element Vector{Float64}:
 -0.14888822835704113
 -0.13399842372338008
 -1.616625283882789
```

In [10]:
```
t = range(-4, length=100, stop=2)
```

Out[10]:
```
-4.0:0.06060606060606061:2.0
```

In [11]:
```
ft = alpha[1]*t.^2 .+ alpha[2]*t .+ alpha[3];
```

In [12]:
```
figure(figsize=(10,4));
scatter(x_k,y_k);
plot(t,ft);
grid();
```



Polynomial of Degree 8

$$f(x) = ax^8 + bx^7 + cx^6 + dx^5 + ex^4 + fx^3 + gx^2 + hx + i$$

Error

$$\varphi(\alpha) = ||A\alpha - B||^2$$

In [13]:
```
A2 = [x_k.^8 x_k.^7 x_k.^6 x_k.^5 x_k.^4 x_k.^3 x_k.^2 x_k ones(9)]
```

Out[13]:
```
9×9 Matrix{Float64}:
 65536.0      -16384.0     …  256.0     -64.0     16.0    -4.0   1.0
 22518.8       -6433.93       150.062   -42.875   12.25   -3.5   1.0
  6561.0       -2187.0         81.0     -27.0      9.0    -3.0   1.0
  1525.88       -610.352       39.0625  -15.625    6.25   -2.5   1.0
     1.0           1.0          1.0       1.0      1.0     1.0   1.0
     4.29982       3.58318  …   2.0736    1.728    1.44    1.2   1.0
```

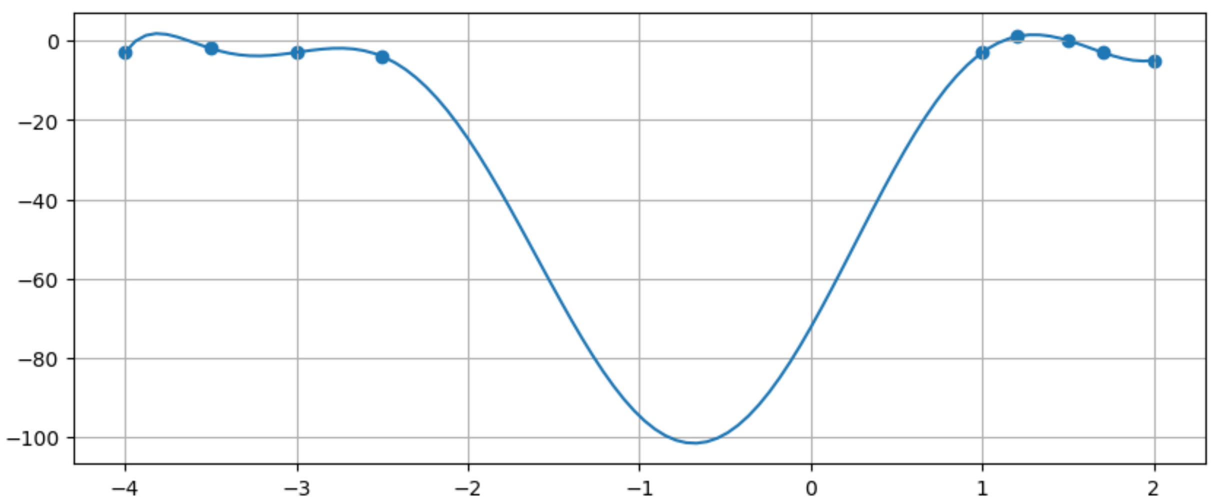
```
 25.6289       17.0859        5.0625    3.375    2.25    1.5   1.0
 69.7576       41.0339        8.3521    4.913    2.89    1.7   1.0
```

In [14]:
```
alpha2 = A2\y_k
```

Out[14]:
```
9-element Vector{Float64}:
  -0.05719715952745426
  -0.2845089613073464З
   1.058510299037152
   5.702255097189326
  -7.11223240893959
 -36.1653531023343
  29.511123503980702
  76.56944425181598
 -72.22204151991448
```

In [15]:
```
ft2 = [t.^8 t.^7 t.^6 t.^5 t.^4 t.^3 t.^2 t ones(100)]*alpha2;
```

In [16]:
```
figure(figsize=(10,4));
scatter(x_k,y_k);
plot(t,ft2);
grid();
```



While the polynomial of degree 8 does go through every point, it would be unlikely that a set of data would include the large dip between x=-2.5 and x=1 that occurs in the polynomial. For that reason the quadratic would be a batter choice for this particular data set

Question 2

$$(E) : ax^2 + bxy + cy^2 + dx + ey + f = 1$$

$$\varphi(\beta) = ||C\beta - D||^2$$

In [1]:
```
x_n =[-5.67;-3.12;-2.05;-2.31;-3.38;3.90;2.69;2.20;3.28;5.83]
```

Out[1]:
```
10-element Vector{Float64}:
```

```
 -5.67
 -3.12
 -2.05
 -2.31
 -3.38
  3.9
  2.69
  2.2
  3.28
  5.83
```

In [2]:

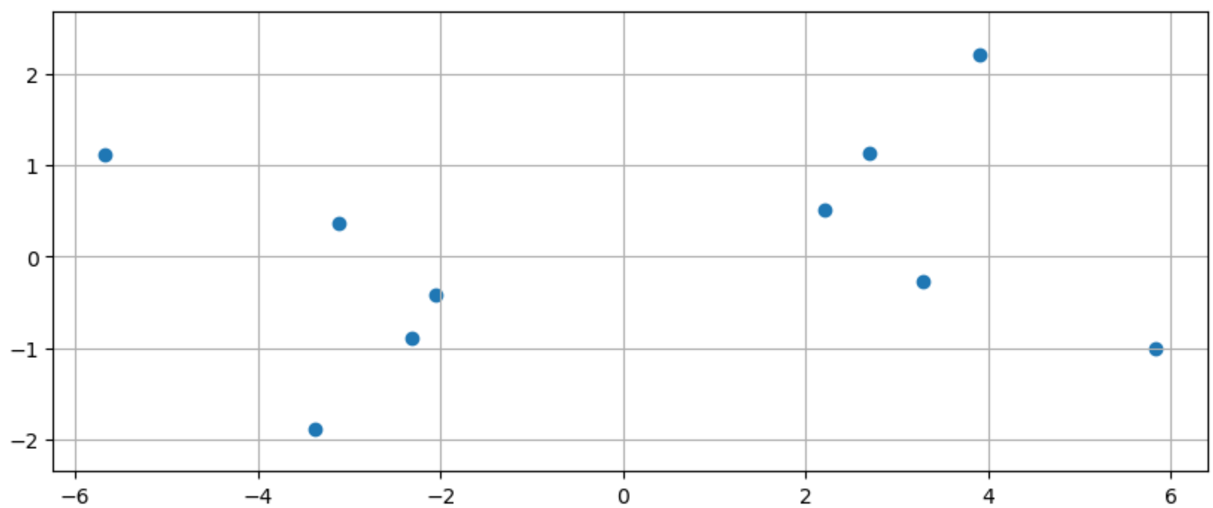
```
y_n =[1.11;0.37;-0.42;-0.89;-1.88;2.21;1.13;0.52;-0.27;-1.01]
```

Out[2]:

```
10-element Vector{Float64}:
  1.11
  0.37
 -0.42
 -0.89
 -1.88
  2.21
  1.13
  0.52
 -0.27
 -1.01
```

In [5]:

```
figure(figsize=(10,4));
scatter(x_n,y_n);
axis("equal");
grid();
```



In [6]:

```
E= [x_n.^2]
```

Out[6]:

```
1-element Vector{Vector{Float64}}:
 [32.1489, 9.7344, 4.2025, 5.3361, 11.424399999999999, 15.209999999999999, 7.2360999999999995, 4.840000000000001, 10.758399999999998, 33.9889]
```

In [7]:

```
C = [x_n.^2 x_n.*y_n y_n.^2 x_n y_n ones(10)]
```

Out[7]:

```
10×6 Matrix{Float64}:
```

```
 32.1489  -6.2937  1.2321  -5.67   1.11  1.0
  9.7344  -1.1544  0.1369  -3.12   0.37  1.0
  4.2025   0.861   0.1764  -2.05  -0.42  1.0
  5.3361   2.0559  0.7921  -2.31  -0.89  1.0
 11.4244   6.3544  3.5344  -3.38  -1.88  1.0
 15.21     8.619   4.8841   3.9    2.21  1.0
  7.2361   3.0397  1.2769   2.69   1.13  1.0
  4.84     1.144   0.2704   2.2    0.52  1.0
 10.7584  -0.8856  0.0729   3.28  -0.27  1.0
```

In [9]:

```
D = ones(10)
```

Out[9]:

```
10-element Vector{Float64}:
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
```

In [10]:

```
beta = C\D
```

Out[10]:

```
6-element Vector{Float64}:
  9.368648627972603e-18
  1.6450873456533273e-17
 -0.0
 -2.0289073398870925e-18
 -7.83393145309818e-18
  0.9999999999999999
```

In [17]:

```
function fimplicit(f,c,xrge,yrge)
n = 1000;
xs = range(xrge[1], stop=xrge[2], length=n);
ys = range(yrge[1], stop=yrge[2], length=n);
xgrid = repeat(xs,1,n);
ygrid = repeat(ys',n,1);
z = f(xgrid,ygrid);
contour(xgrid, ygrid, z, levels=c);
end
```

Out[17]:

```
fimplicit (generic function with 1 method)
```

In [13]:

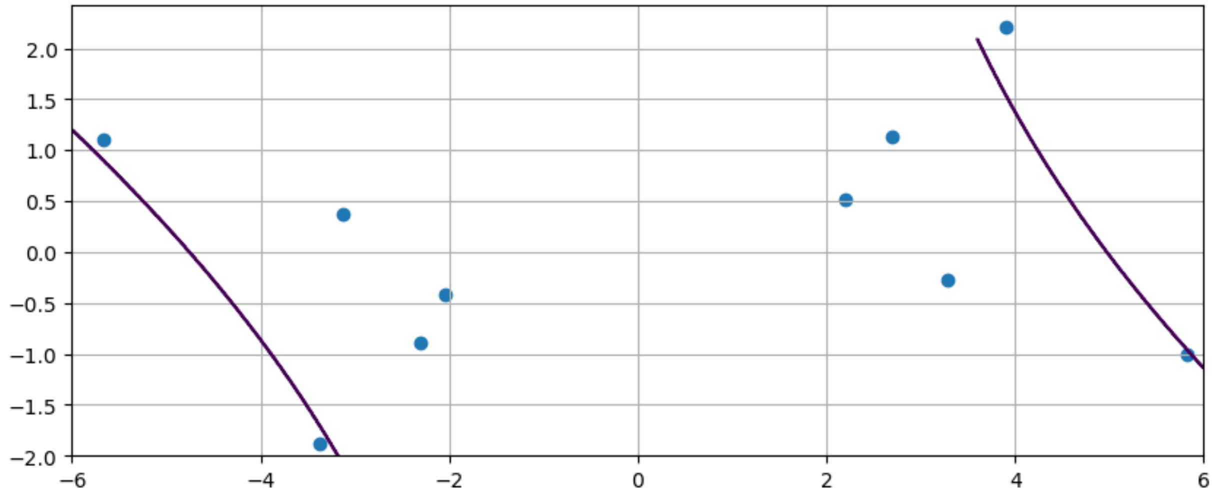
```
f = (x,y) -> beta[1]*x.^2 .+  beta[2]*x.*y .+  beta[3]*y.^2 .+ beta[4]x .+ bet
```

Out[13]:

```
#1 (generic function with 1 method)
```

In [18]:

```
figure(figsize=(10,4));
scatter(x_n,y_n);
fimplicit(f,[1],[-6,6],[-2,2.1]);
grid();
```



This doens't look close...but math seems correct

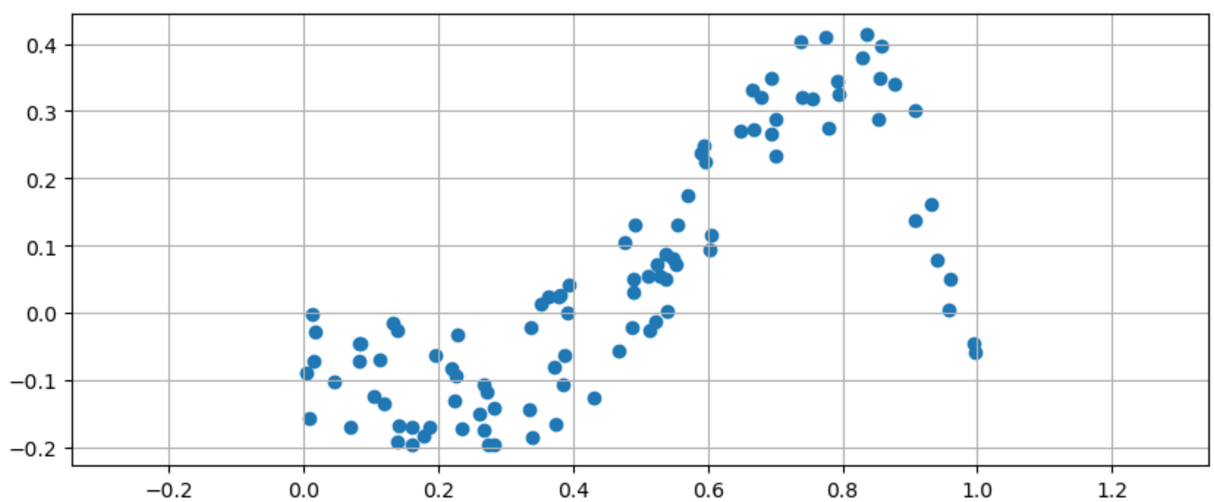
Question 3

In [27]:

```
n=100;
t=sort(rand(n));
b=(t.-0.3).*sin.(pi*t.^2)-0.2*rand(n);
```

In [28]:

```
figure(figsize=(10,4));
scatter(t,b);
axis("equal");
grid();
```



$$min_{x \in R^n} ||x - b||^2 + \lambda R(x)$$

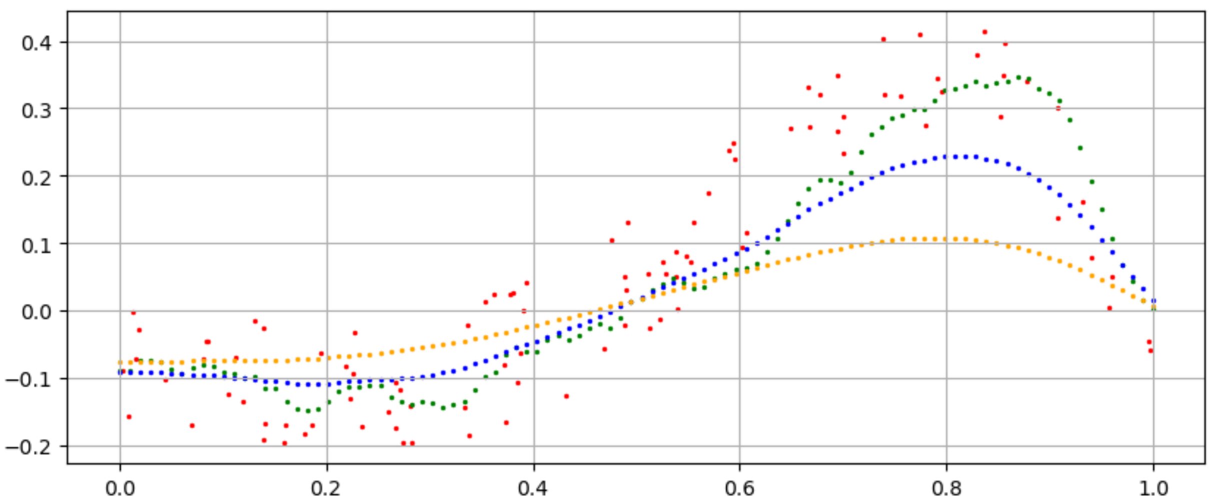
$$D(x) = \sum^{n-1}(x \quad x \quad)^2$$

In [29]:
```
D = diagm(0 => ones(n), 1 =>-ones(n));
```

In [30]:
```
D = D[1:end-1, 1:end-1];
```

In [31]:
```
lambda = 5;
xLS = (I+lambda*D'*D)\b;
lambda = 100;
xLS2 = (I+lambda*D'*D)\b;
lambda = 500;
xLS3 = (I+lambda*D'*D)\b;
t_norm = range(0,stop=1,length=100);
```

In [32]:
```
figure(figsize=(10,4));
grid();
#plot(t,x);
scatter(t,b, s=2, c="red");
scatter(t_norm,xLS, s=2, c="green");
scatter(t_norm,xLS2, s=2, c="#0000FF");
scatter(t_norm,xLS3, s=2, c="Orange");
```



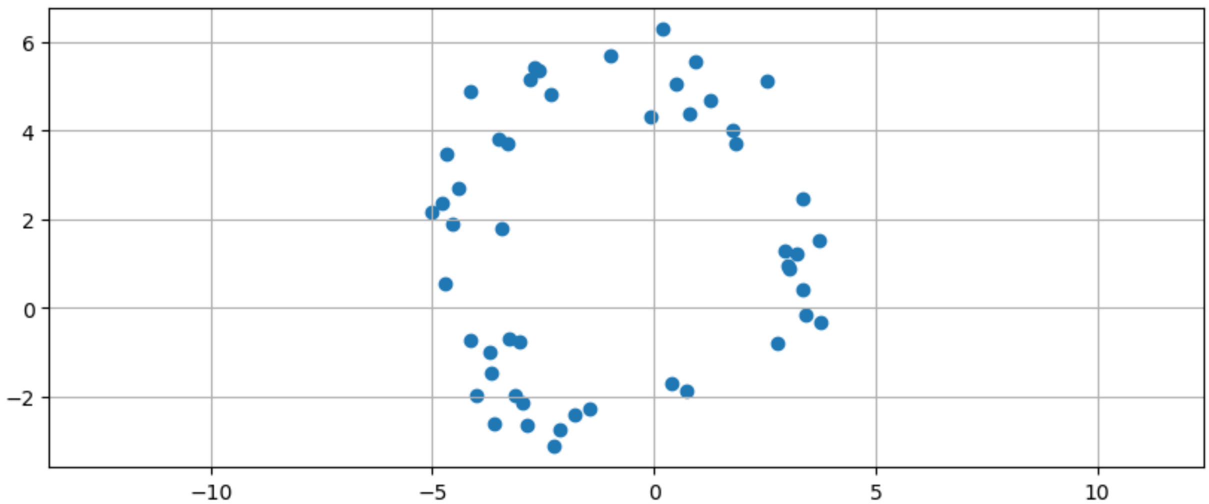
Question 4

In [33]:
```
m=50;
alpha = -rand(m);
beta = rand(m) + ones(m);
eta = 2*rand(m) .+ 3*ones(m);
theta = 2pi*rand(m);
```

$$(u_i, v_i) = (\alpha_i + \eta_i cos\theta_i + \eta_n sin\theta_i)$$

In [34]:
```
u = alpha .+ eta.*cos.(theta);
v = beta  .+ eta.*sin.(theta);
```

In [35]:
```
figure(figsize=(10,4));
scatter(u,v);
axis("equal");
grid();
```



In [36]:
```
uv = [u v];
```

In [37]:
```
A = [-2uv ones(m)];
```

In [38]:
```
y = -u.^2 -v.^2;
```

In [39]:
```
xhat = A\y
```

Out[39]:
```
3-element Vector{Float64}:
  -0.7386221428035423
   1.4094762188120862
 -14.122480438466958
```

Center

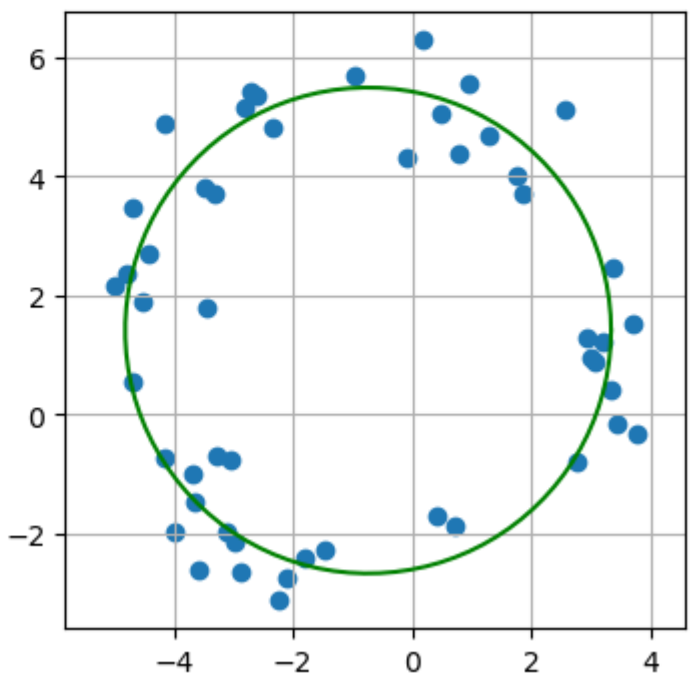
In [40]:
```
x = xhat[1:end-1]
```

Out[40]:
```
2-element Vector{Float64}:
 -0.7386221428035423
  1.4094762188120862
```

Radius

In [41]:
```
R = xhat[end]
```
Out[41]:
```
-14.122480438466958
```
In [42]:
```
r = sqrt(norm(x)^2-R)
```
Out[42]:
```
4.081012903643343
```

Graph

In [43]:
```
theta_graph =  range(0, stop=2pi, length=100);
```
In [44]:
```
cx = x[1] .+ r*cos.(theta_graph);
cy = x[2] .+ r*sin.(theta_graph);
```
In [45]:
```
figure(figsize=(4,4));
grid();
axis("equal");
scatter(u, v);
plot(cx,cy, "g");
```


In [ ]:
```

```