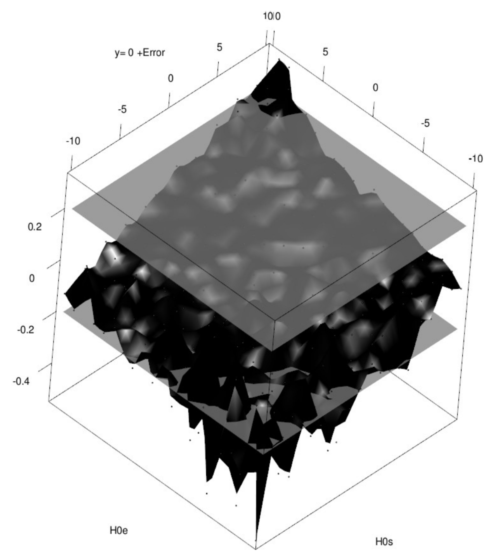
Let's take a sample $s \in Uniform(min=0, entropy=H0_s)$ and a related sample $y = f(s) + Normal(mean=0, entropy=H0_e)$ where *f(s)* is a function of *s*. Then we can estimate the entropies: *ebc_sample(s)~*H_x, *ebc_sample(y)~*H_y, *ebc_sample2d(x,y)~*H_{xy} and $I=H_x+H_y-H_{xy}$.. This could be done with the *explore_I* function in *ebc.R*[1]. This function estimates *I,* H_x, H_y,H_{xy} for each $(H0_s,H0_e)$, plots *I vs.* $(H0_s,H0_e)$ and returns a *data.frame* with the values *{R²,H0s,H0e,Hx,Hy,Hxy.I}*

In the graph the points are colored according the absolute value of *adjusted R²* from the linear model *lm(y~s)*.

color	$\lvert R^2 \rvert \in (a,b)$
Black	(0 , 0.25)
red	(0.25 , 0.5)
green	(0.5 , 0.75)
blue	(0.75 , 1)

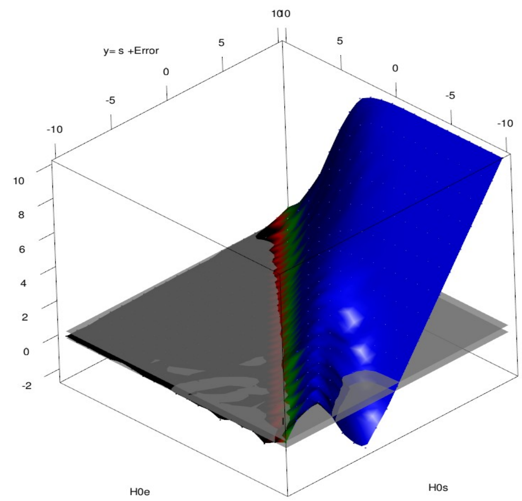
Decoration: two planes at *I={-0.2,0.2}*

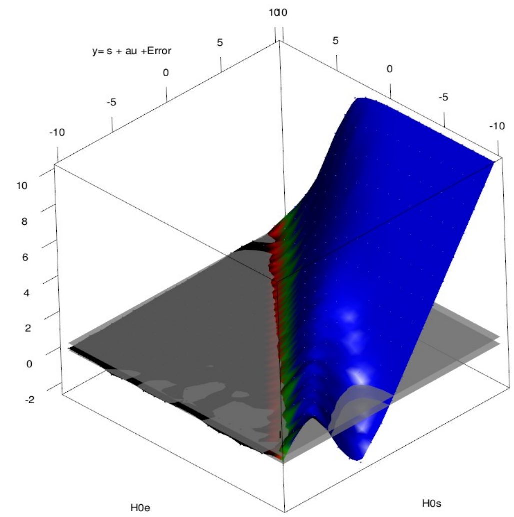
Two random samples, one uniform and normal the other. *f(x)=0*.

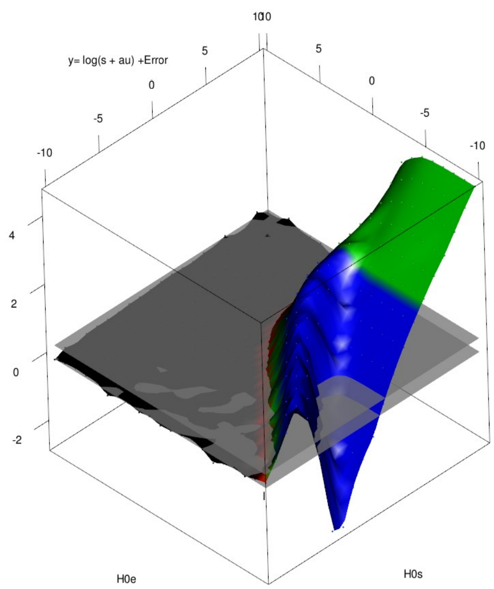


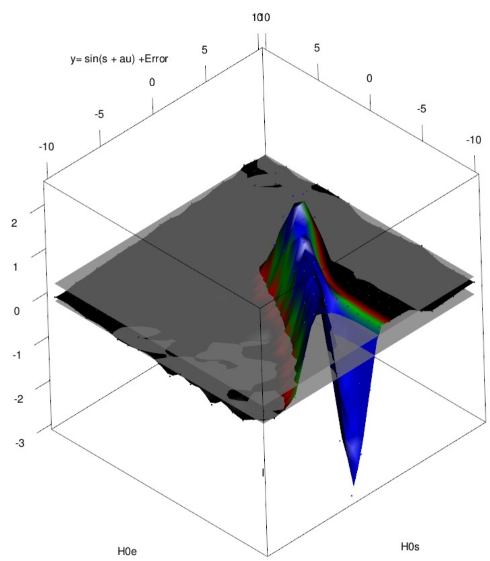
1 https://docs.google.com/file/d/0B6ZuqpeSKSqcamlrWDdKWlJQWTg/edit?usp=sharing

f(s)=s and *f(s)=s+au*. Where *au* is the golden number $\varphi = \frac{1+\sqrt{(5)}}{2}$ (this is computed inside *explore_1*)



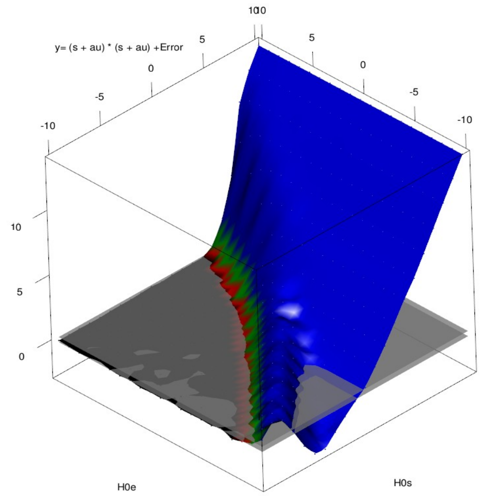


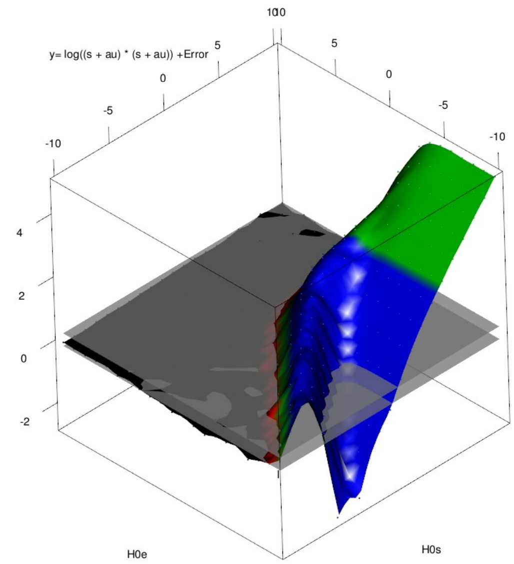


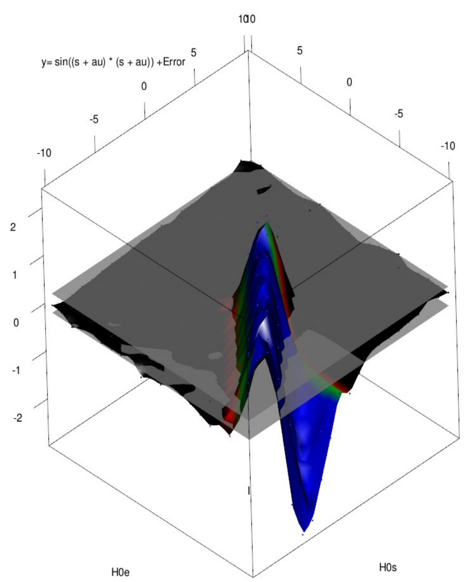


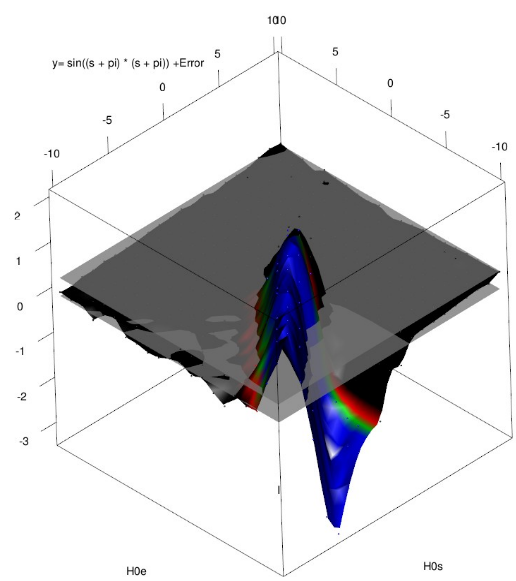
f(s)=(s+au)(s+au).

```
explore_I(func=expression((s+au)*(s+au)),okplot=T,H_ref=c(-10,10),npts=
20,N=1e4)
```

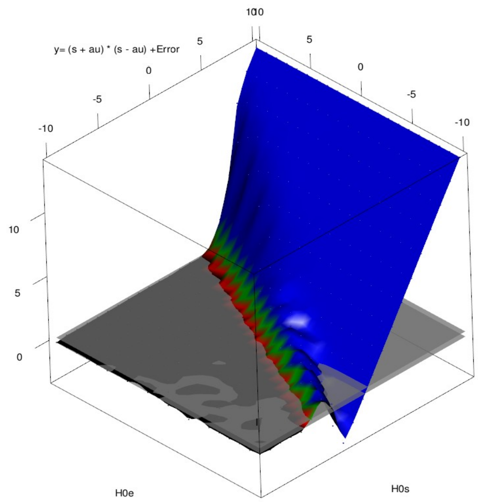


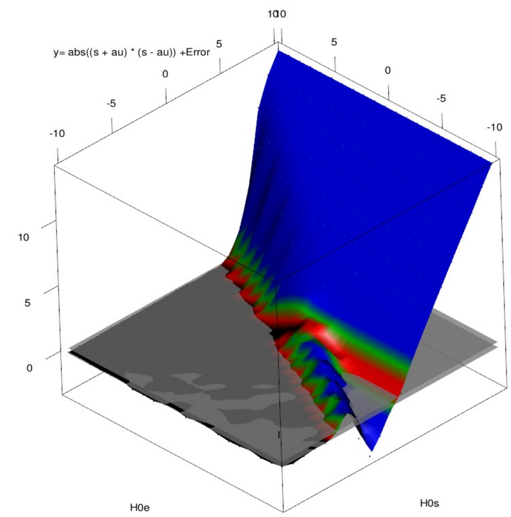


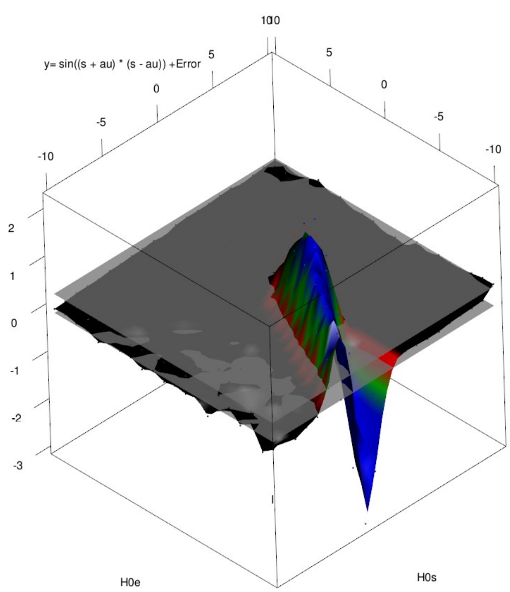


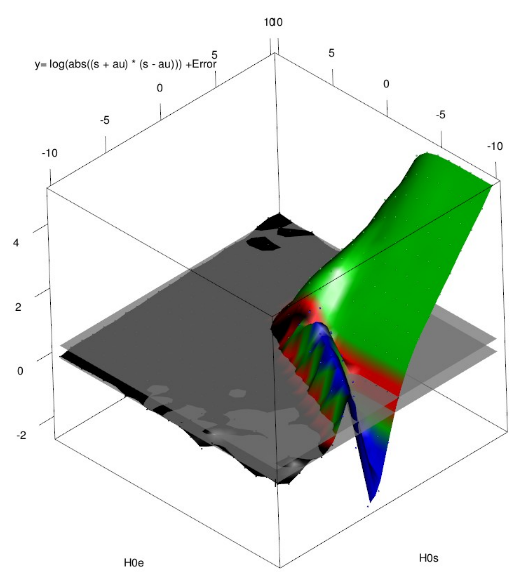


f(x)=(x+au)(x-au).

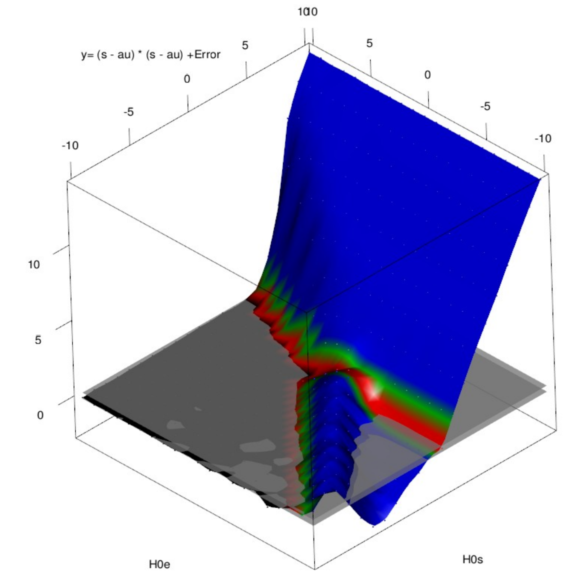

y= (s + au) * (s - au) +Error
H0e
H0s

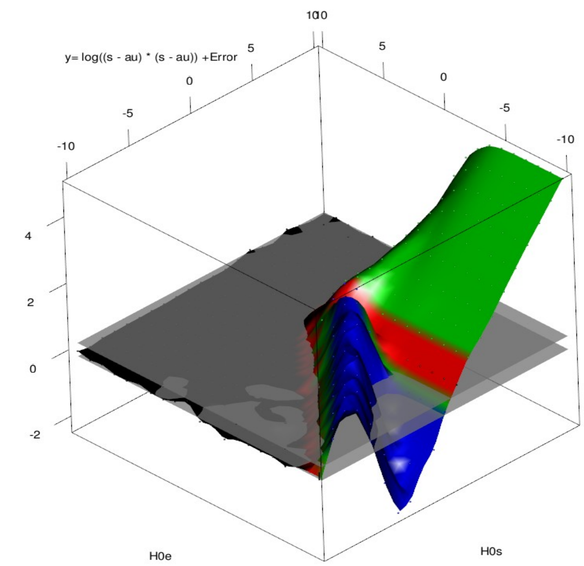

y= abs((s + au) * (s - au)) +Error
H0e
H0s

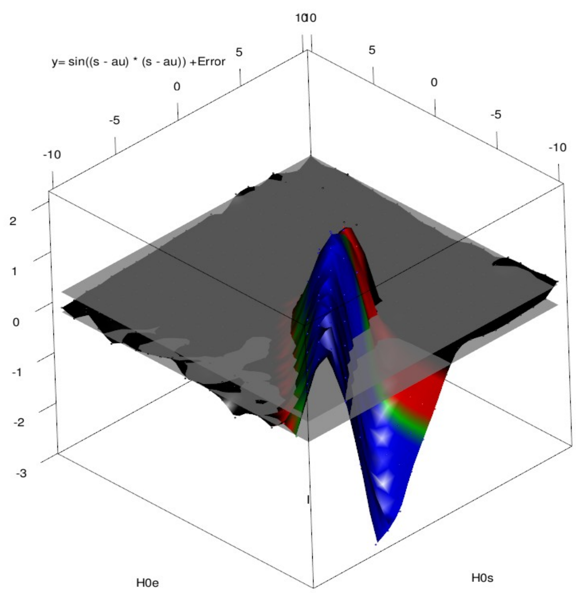

y= sin((s + au) * (s - au)) +Error
H0e
H0s

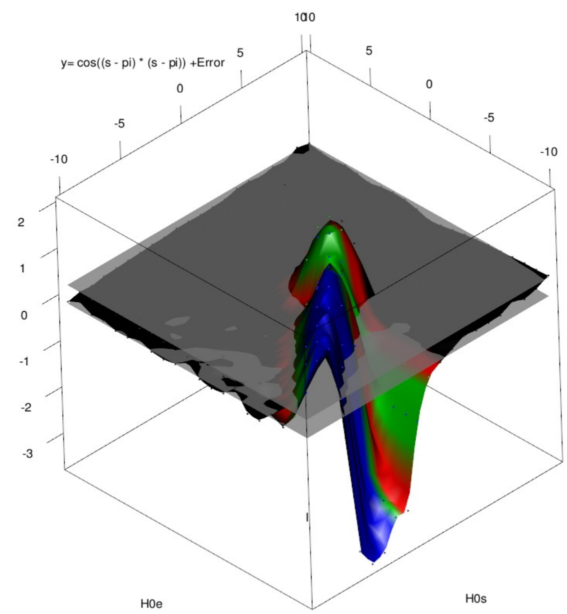

y= log(abs((s + au) * (s - au))) +Error
H0e
H0s

$f(x)=(x\text{-}au)(x\text{-}au)$.

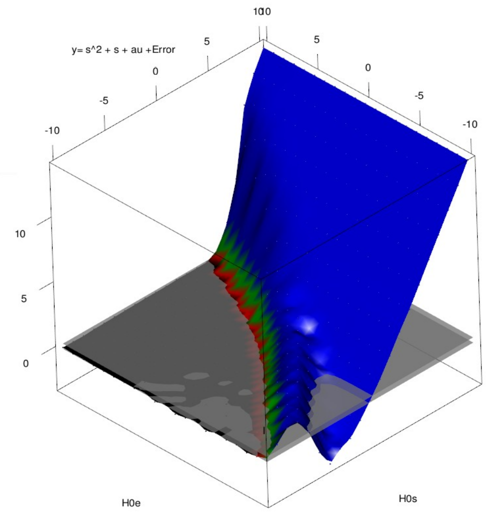


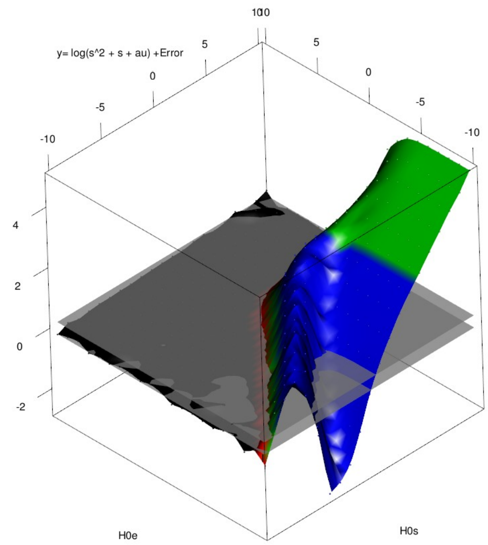


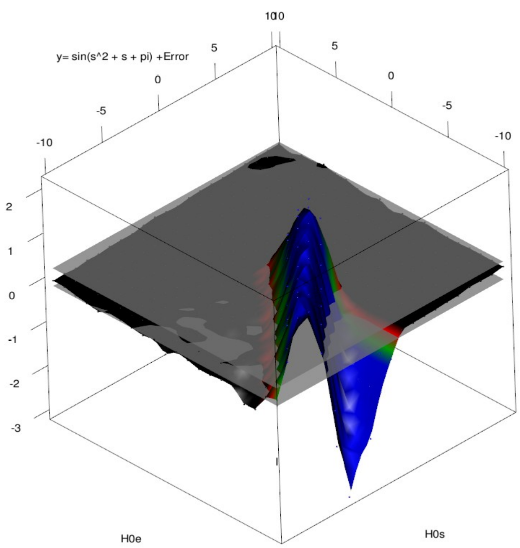


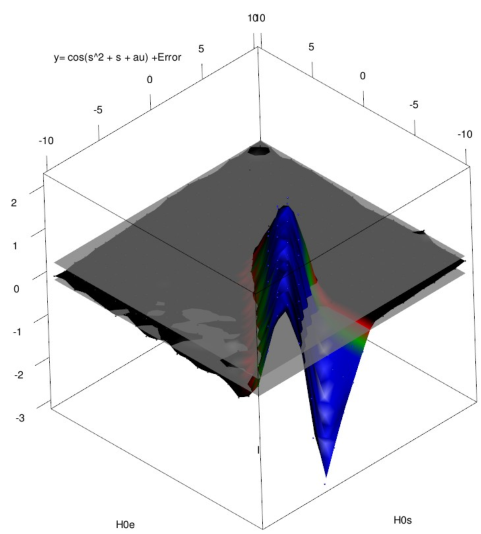


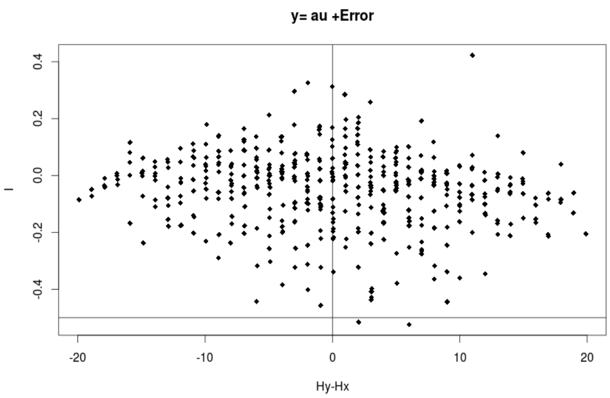
$f(x)=(x^2+x+au)$.

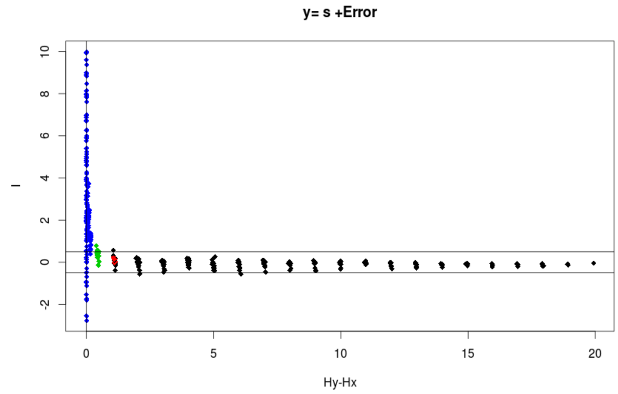

y= s^2 + s + au +Error
H0e
H0s

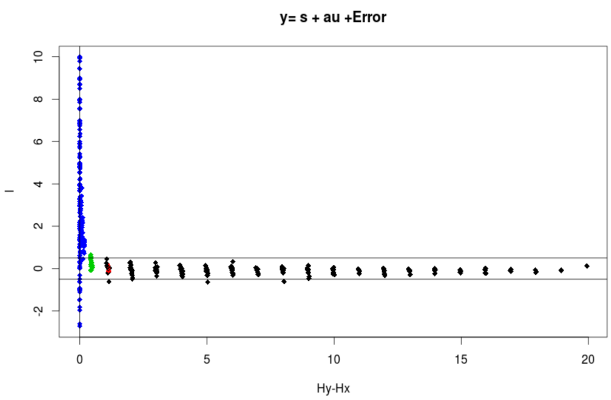

y= log(s^2 + s + au) +Error
H0e
H0s

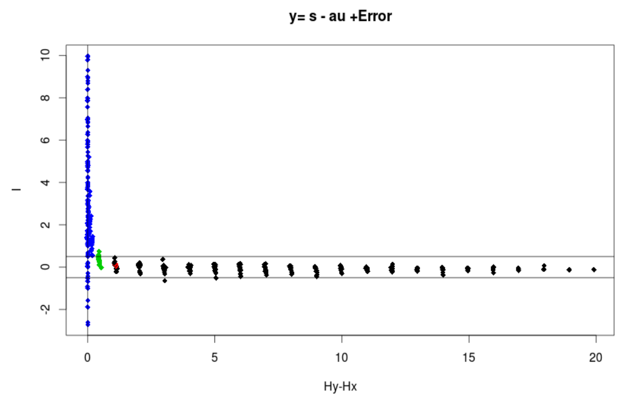

y= sin(s^2 + s + pi) +Error
H0e
H0s

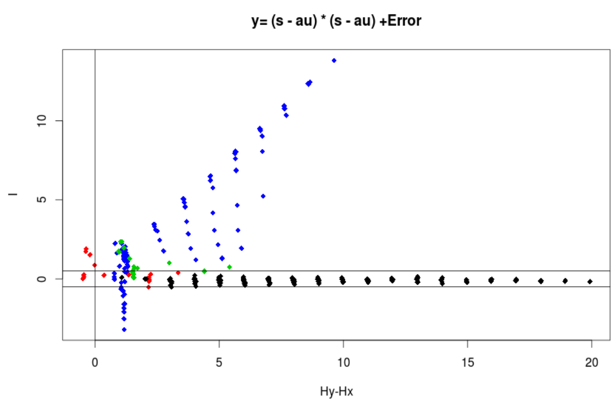

y= cos(s^2 + s + au) +Error
H0e
H0s

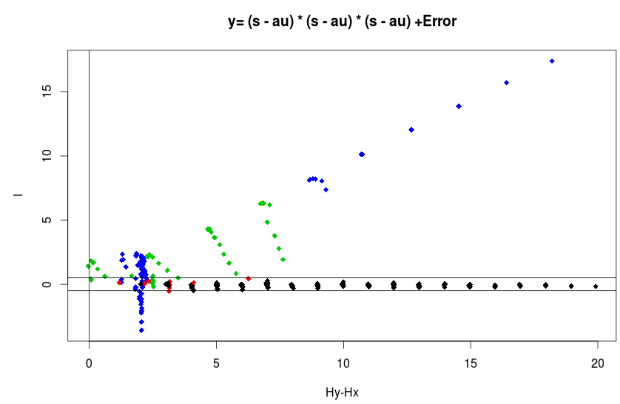

y= au +Error
Hy-Hx
I

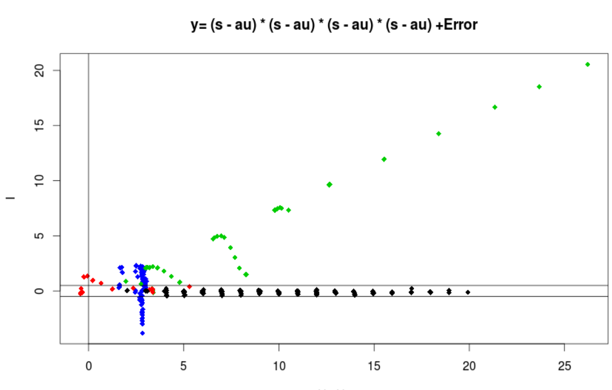

y= s +Error
Hy-Hx
I

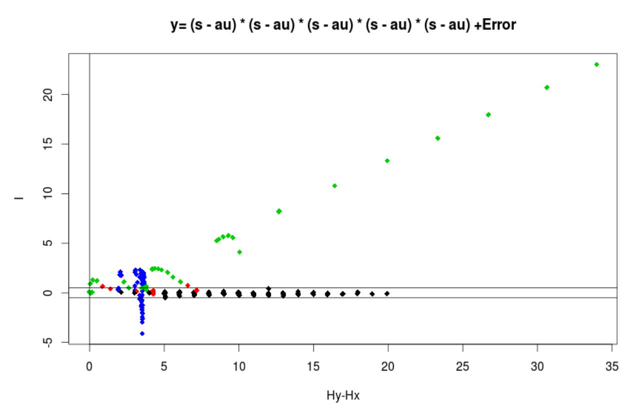

y= s + au +Error
Hy-Hx
I

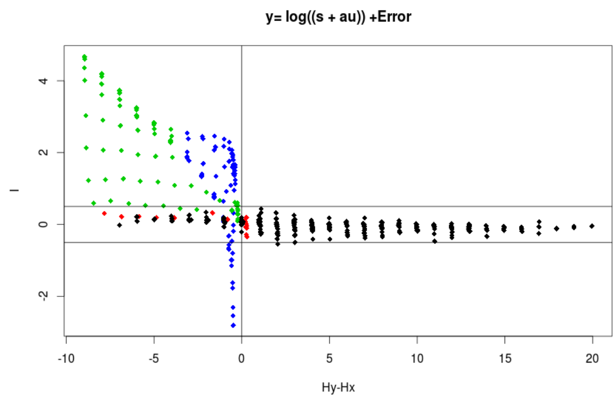

y= s - au +Error
Hy-Hx
I

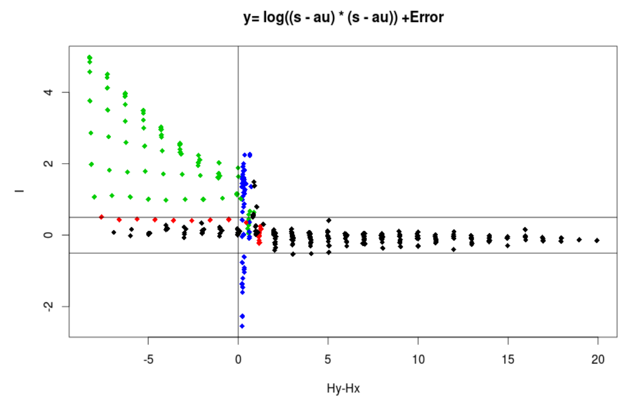

y= (s - au) * (s - au) +Error
Hy-Hx
I

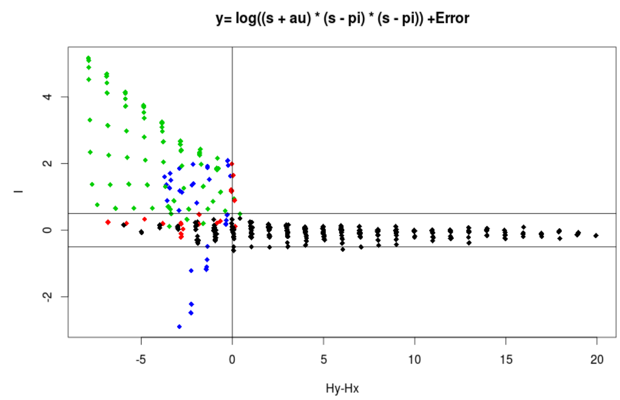

y= (s - au) * (s - au) * (s - au) +Error
Hy-Hx
I

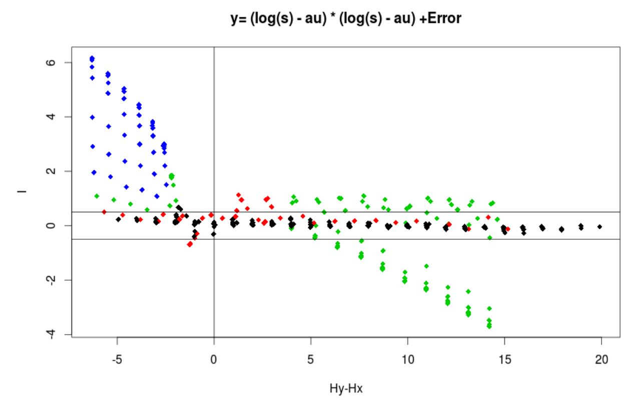

y= (s - au) * (s - au) * (s - au) * (s - au) +Error
Hy-Hx
I

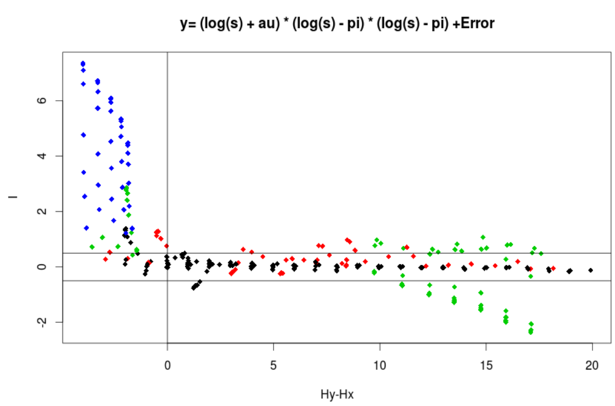

y= (s - au) * (s - au) * (s - au) * (s - au) * (s - au) +Error
Hy-Hx
I

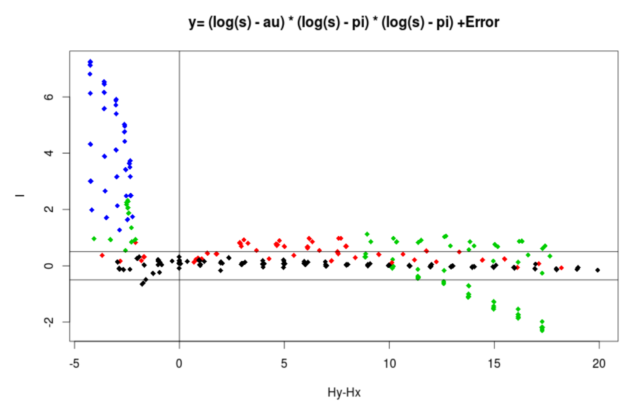

y= log((s + au)) +Error
Hy-Hx
I

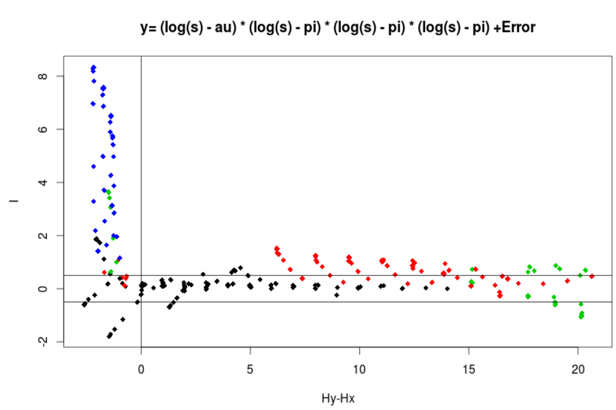

y= log((s - au) * (s - au)) +Error
Hy-Hx
I

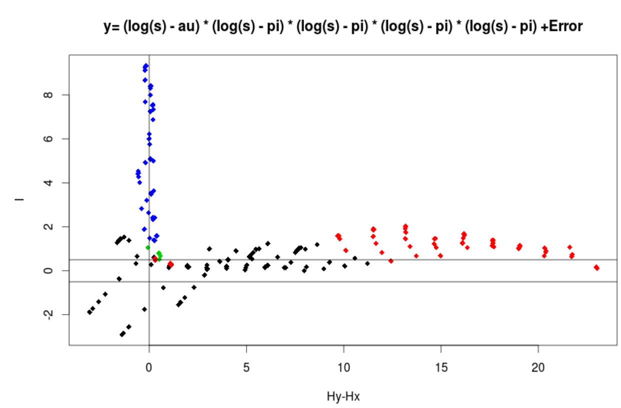

y= log((s + au) * (s - pi) * (s - pi)) +Error
Hy-Hx
I


y= (log(s) - au) * (log(s) - au) +Error
Hy-Hx
I

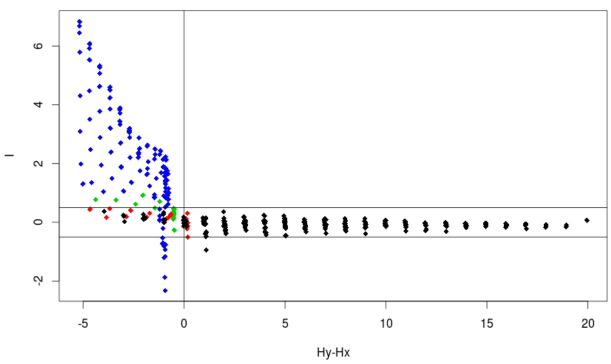

y= (log(s) + au) * (log(s) - pi) * (log(s) - pi) +Error
Hy-Hx
I


y= (log(s) - au) * (log(s) - pi) * (log(s) - pi) +Error
Hy-Hx
I

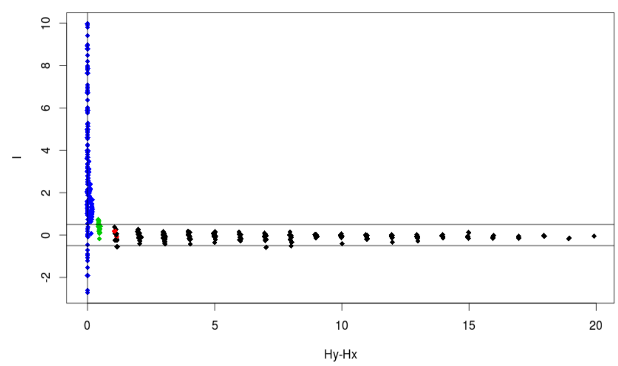

y= (log(s) - au) * (log(s) - pi) * (log(s) - pi) * (log(s) - pi) +Error
Hy-Hx
I


y= (log(s) - au) * (log(s) - pi) * (log(s) - pi) * (log(s) - pi) * (log(s) - pi) +Error
Hy-Hx
I

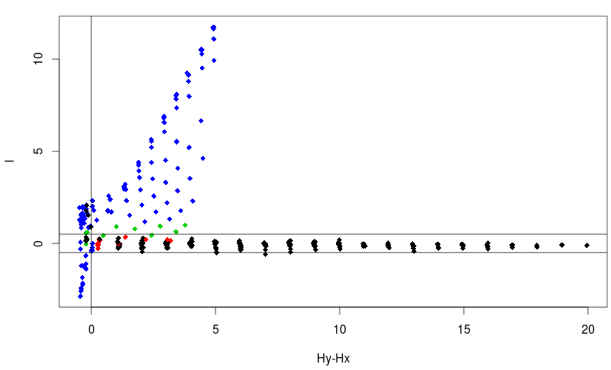
y= sqrt((s + au)) +Error



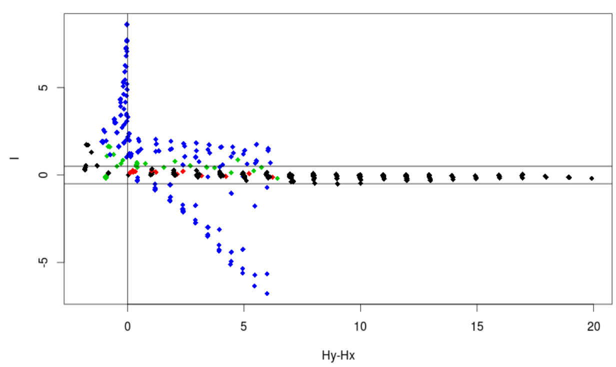
y= sqrt((s + au) * (s + au)) +Error



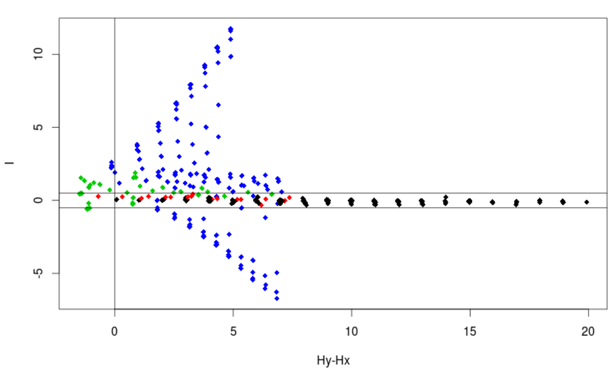
y= sqrt((s + au) * (s - au) * (s - au)) +Error



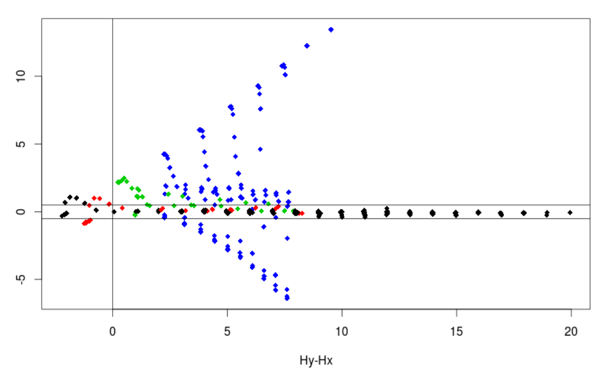
y= (sqrt(s) - au) * (sqrt(s) - au) +Error



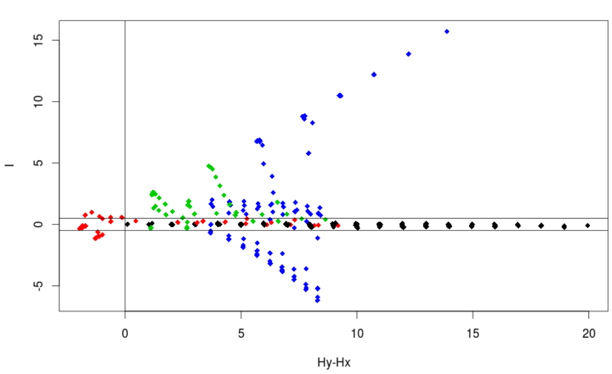
y= (sqrt(s) - au) * (sqrt(s) - au) * (sqrt(s) - au) +Error



y= (sqrt(s) - au) * (sqrt(s) - au) * (sqrt(s) - au) * (sqrt(s) - au) +Error



y= (sqrt(s) - au) * (sqrt(s) - au) * (sqrt(s) - au) * (sqrt(s) - au) * (sqrt(s) - au) +Error



y= (sqrt(s) + au) * (sqrt(s) + au) * (sqrt(s) - au) * (sqrt(s) - au) * (sqrt(s) - au) +Error