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}*

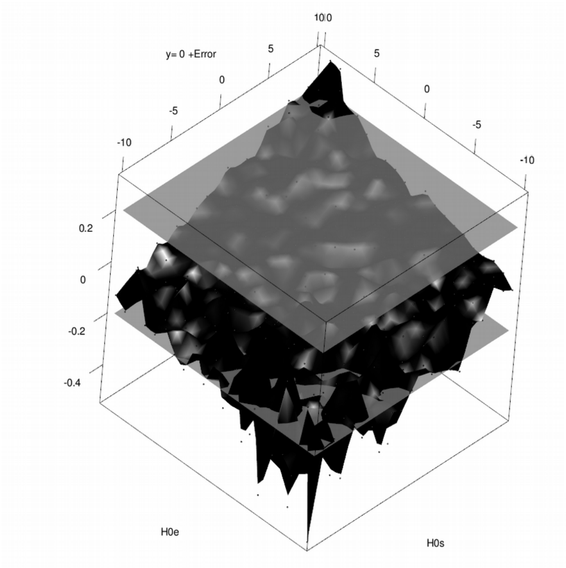
H0	Uniform (max-min)	Normal (σ)
-10	4.5e-5	1e-5
10	2.2e4	5e3

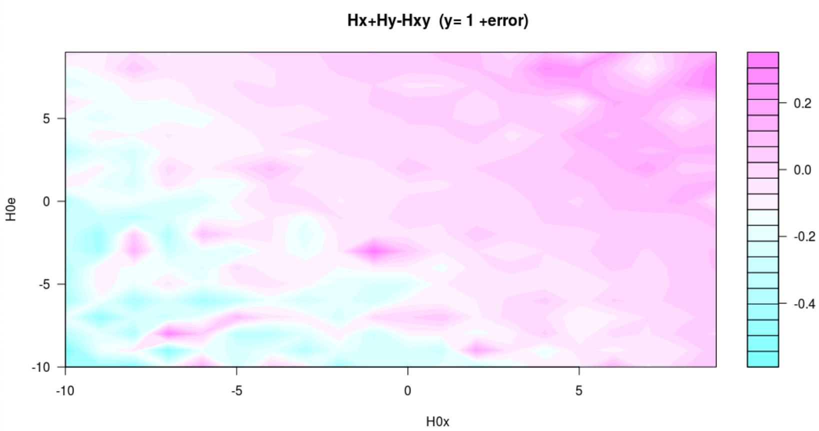
In the surface graph the points are colored according the absolute value of *adjusted R²* from the model *lm(y~f(s))*.

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)

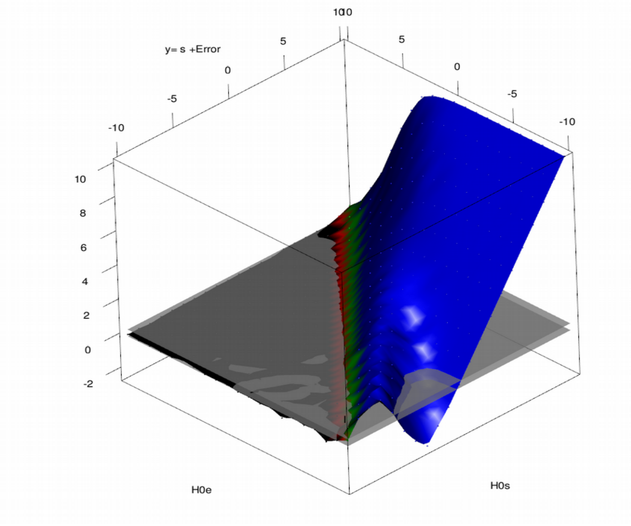
Decoration: two planes at *I={-0.2,0.2}*

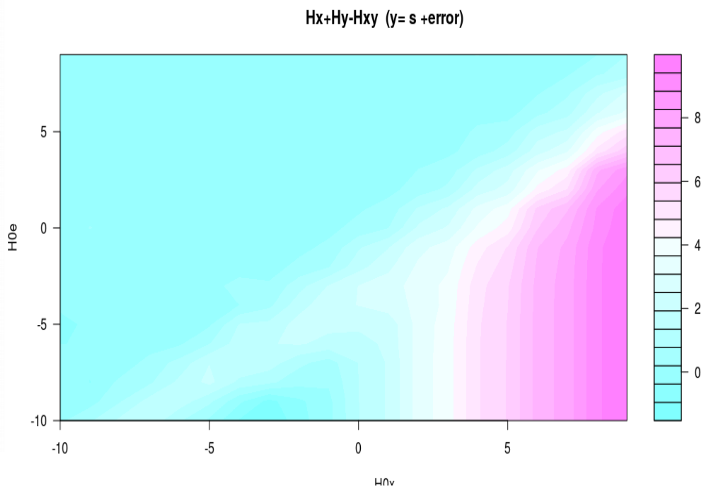
Two random samples, one uniform and normal the other. *f(x)=1*.

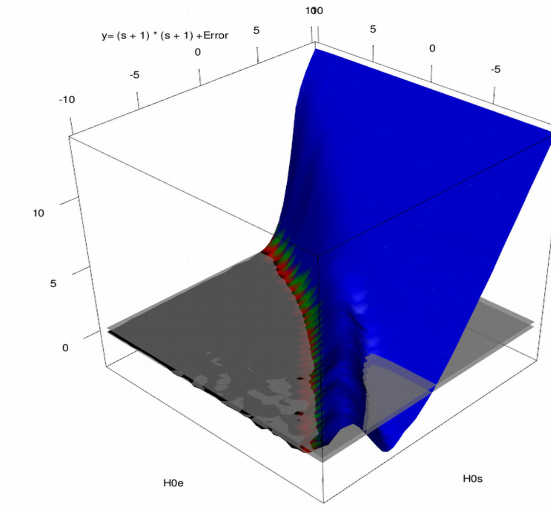


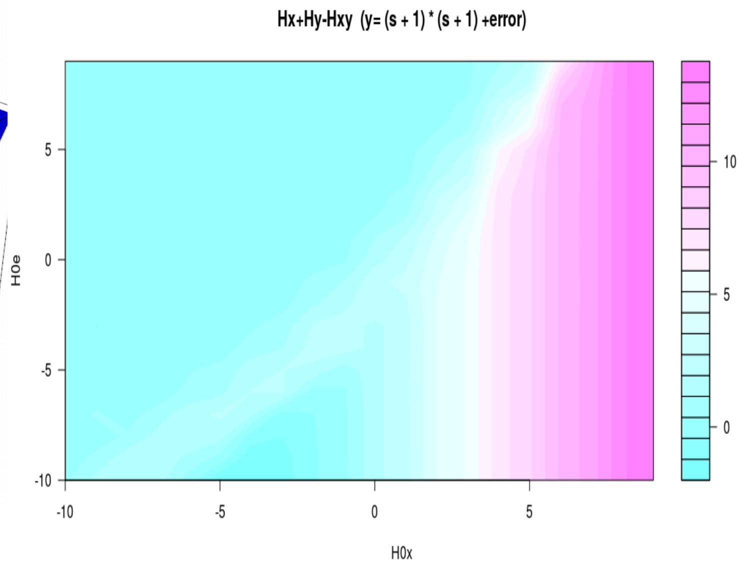


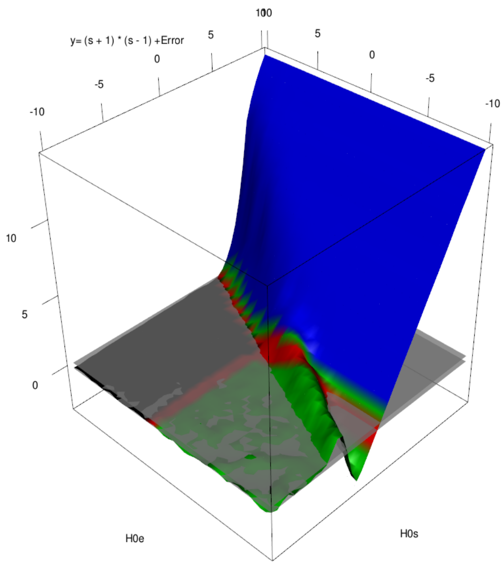
1 https://github.com/llorenzo62/Entropy

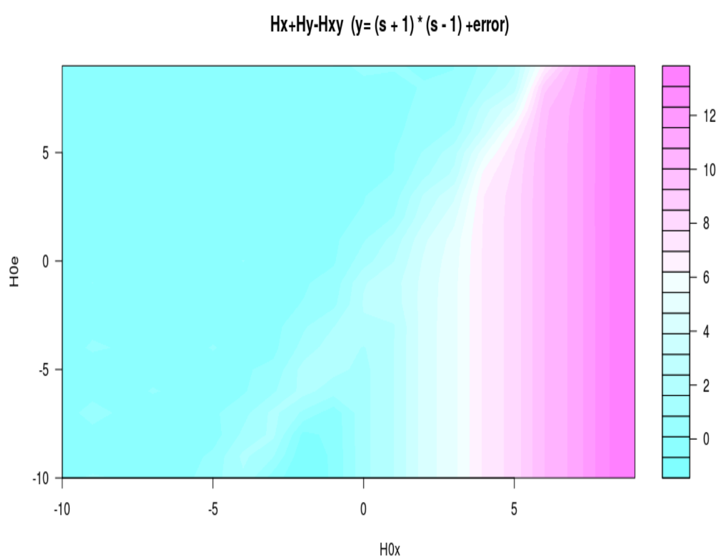

y= s +Error
H0e
H0s

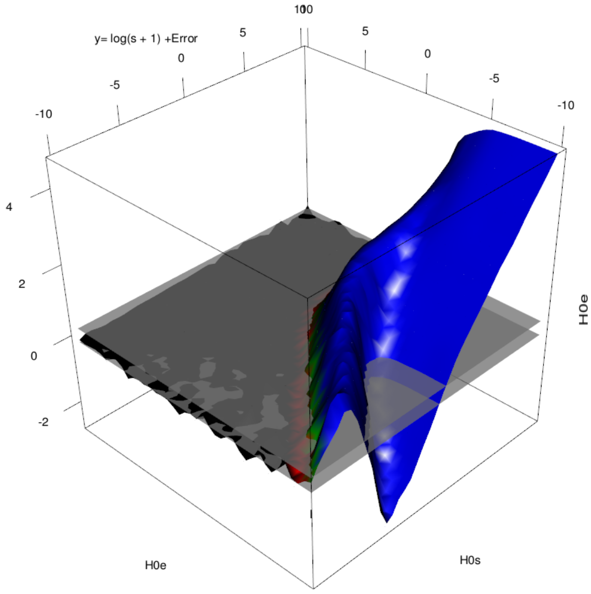

Hx+Hy-Hxy (y= s +error)
H0e
H0x

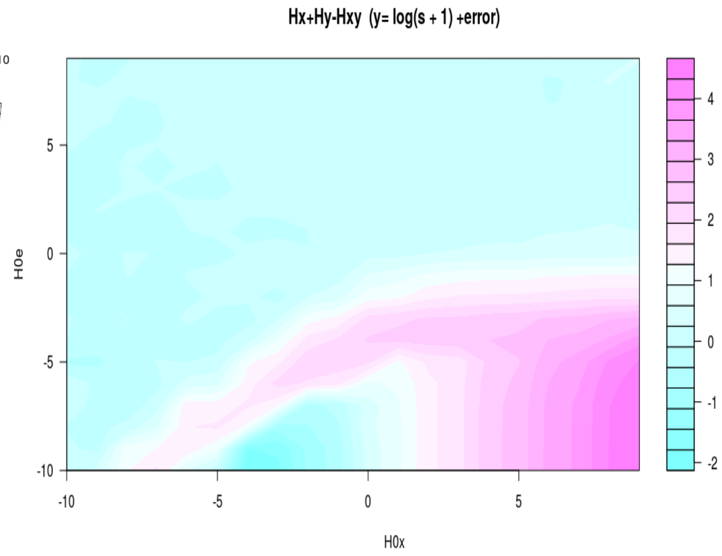

y= (s + 1) * (s + 1) +Error
H0e
H0s

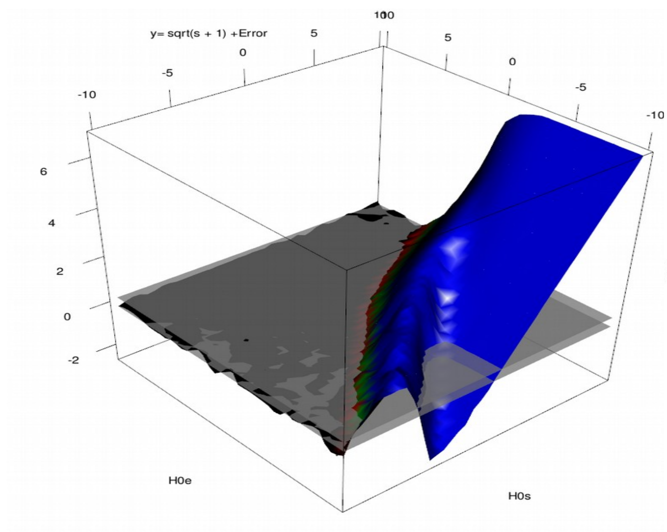

Hx+Hy-Hxy (y= (s + 1) * (s + 1) +error)
H0e
H0x

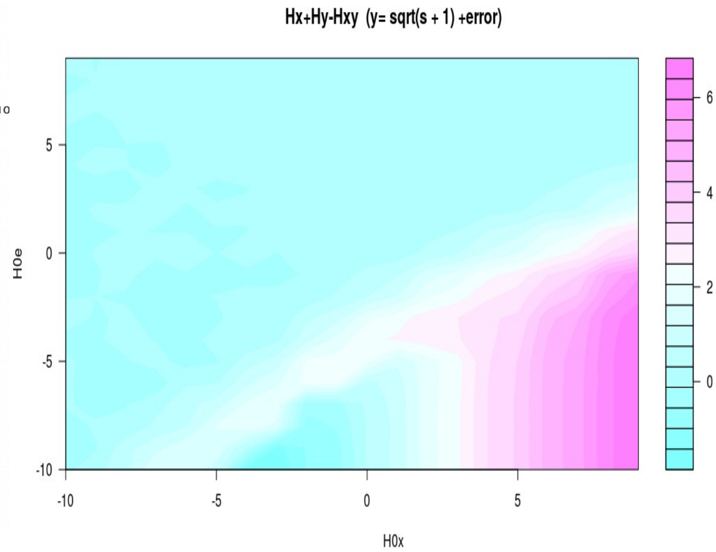

y= (s + 1) * (s - 1) +Error
H0e
H0s

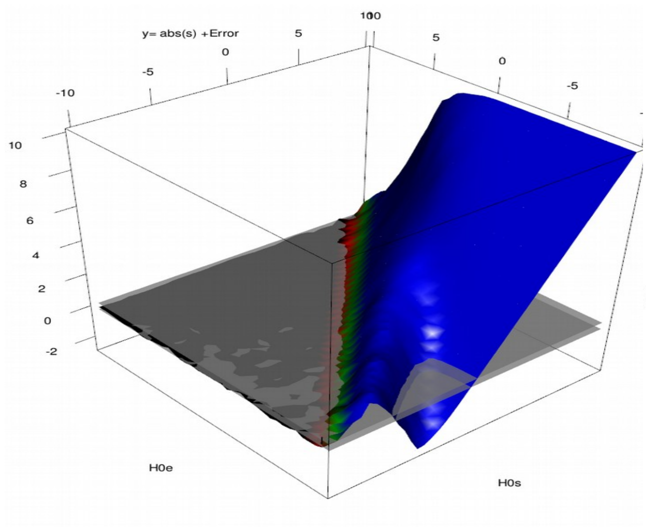

Hx+Hy-Hxy (y= (s + 1) * (s - 1) +error)
H0e
H0x

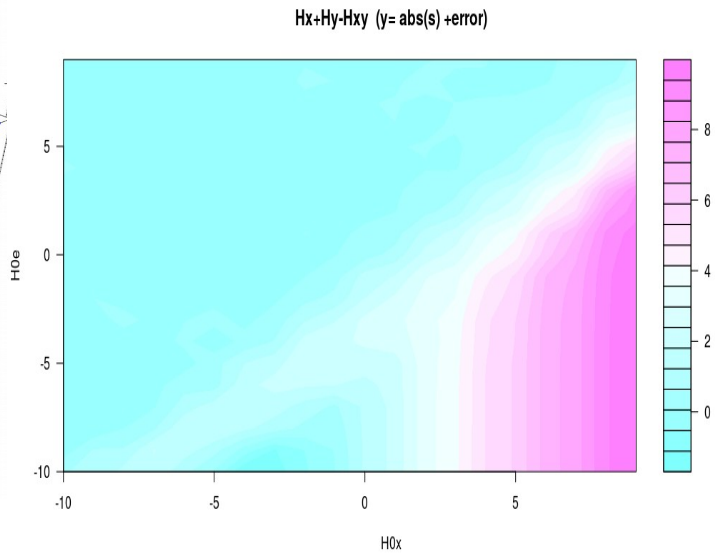

y= log(s + 1) +Error
H0e
H0s

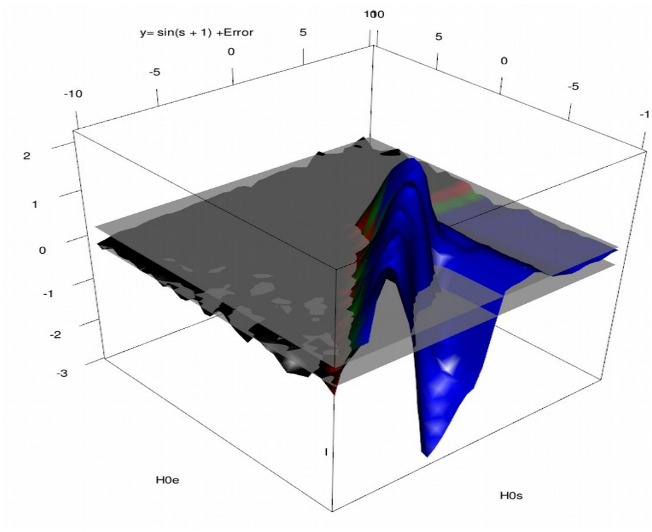

Hx+Hy-Hxy (y= log(s + 1) +error)
H0e
H0x

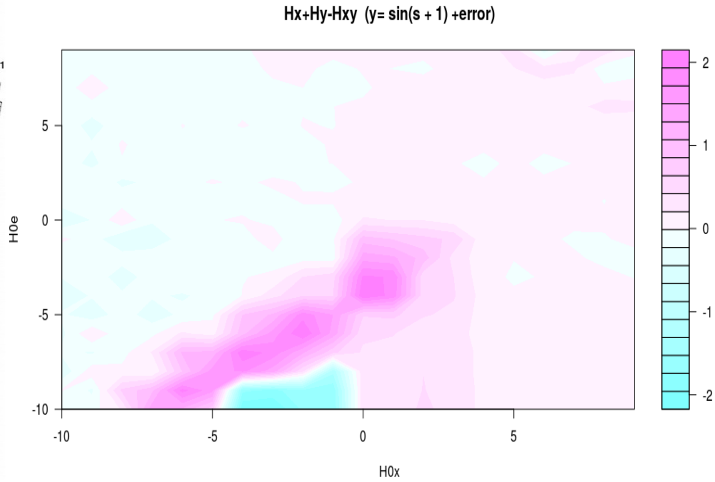

y= sqrt(s + 1) +Error
H0e
H0s

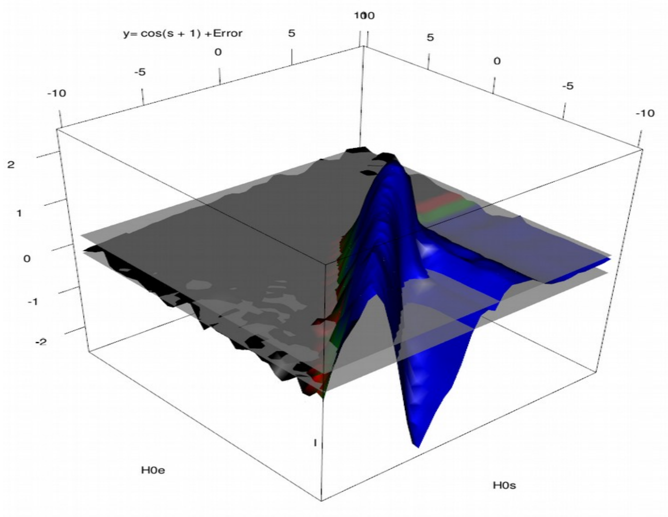

Hx+Hy-Hxy (y= sqrt(s + 1) +error)
H0e
H0x

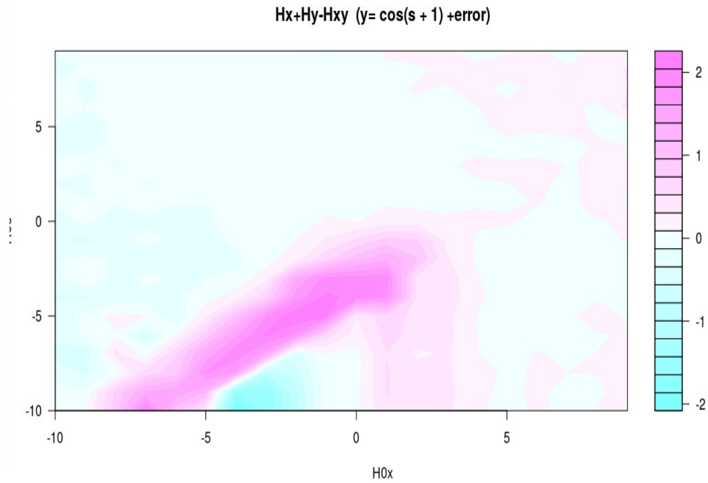

y= abs(s) +Error
H0e
H0s

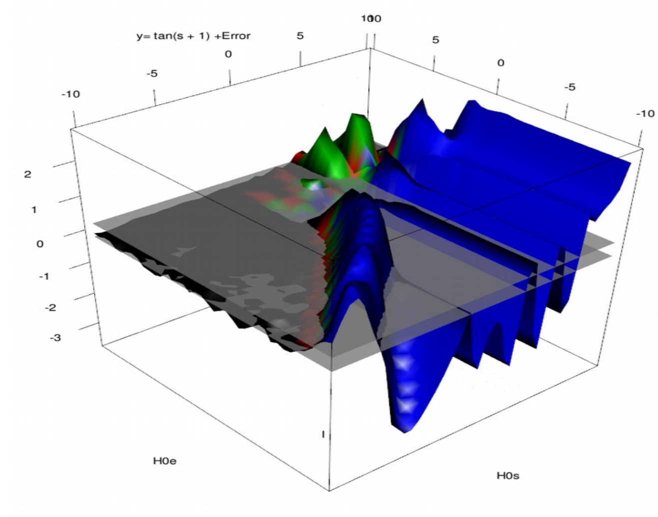

Hx+Hy-Hxy (y= abs(s) +error)
H0e
H0x

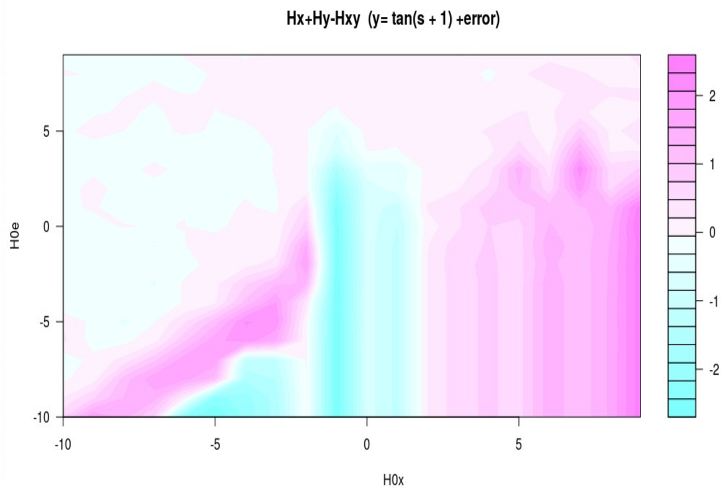

y= sin(s + 1) +Error
H0e
H0s

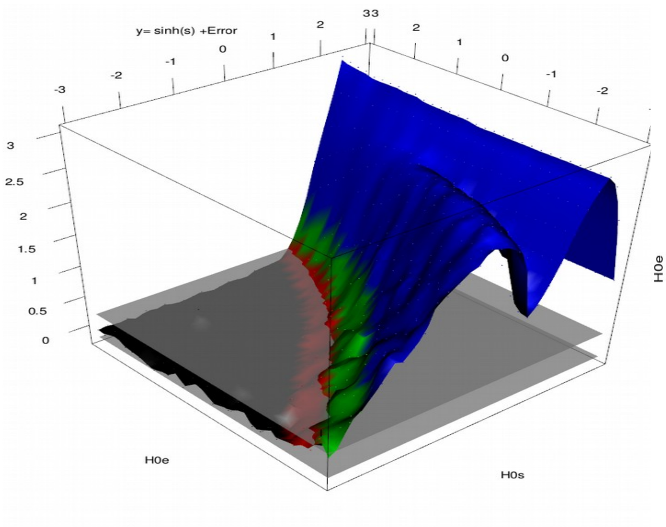

Hx+Hy-Hxy (y= sin(s + 1) +error)
H0e
H0x

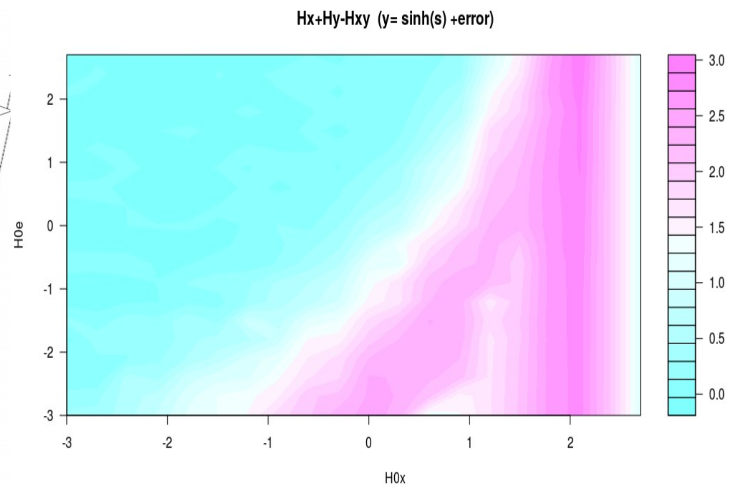

y= cos(s + 1) +Error
H0e
H0s

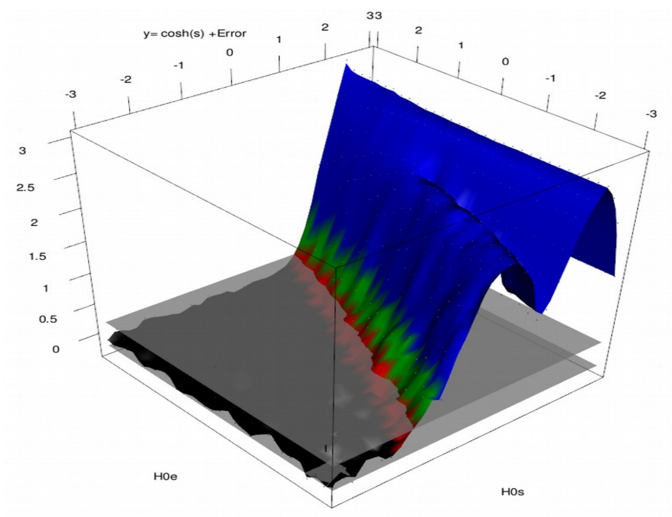

Hx+Hy-Hxy (y= cos(s + 1) +error)
H0x

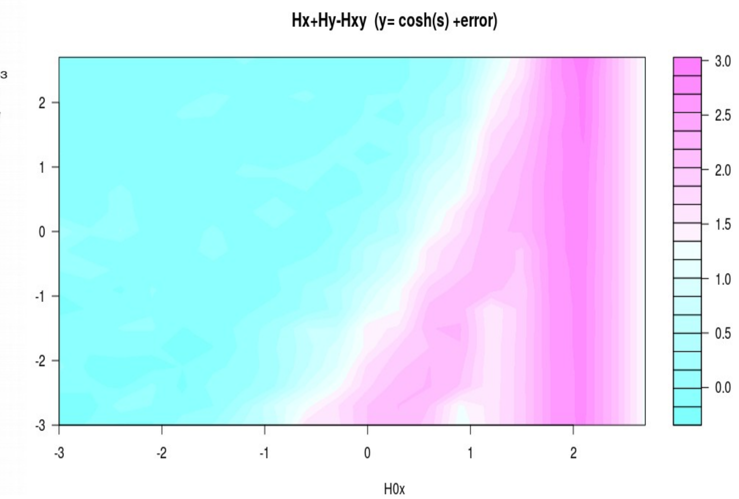

y= tan(s + 1) +Error
H0e
H0s

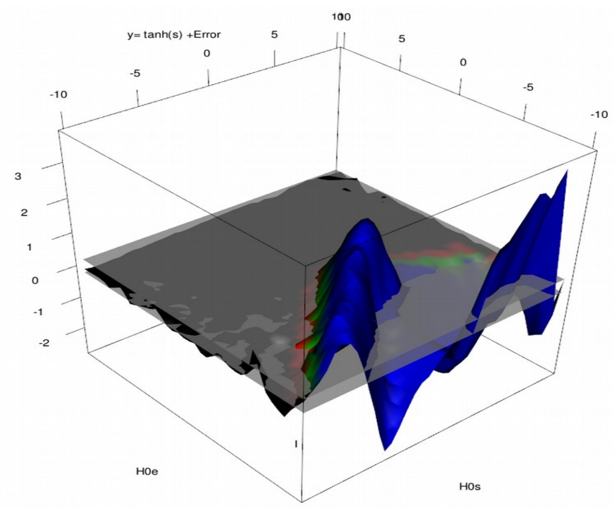

Hx+Hy-Hxy (y= tan(s + 1) +error)
H0e
H0x

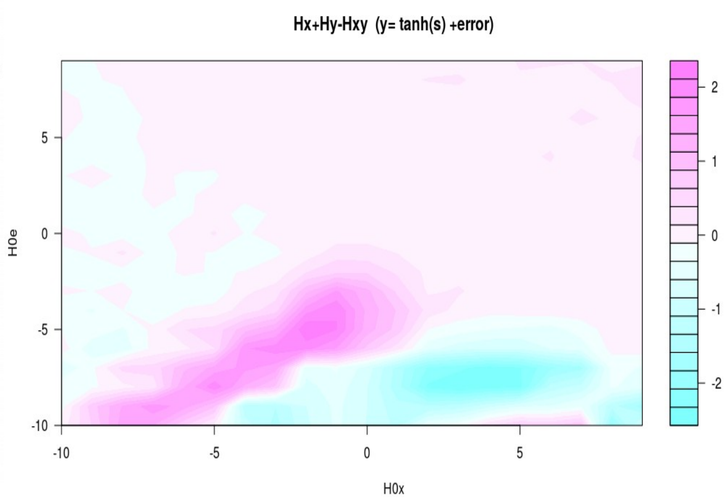

y= sinh(s) +Error
H0e
H0s

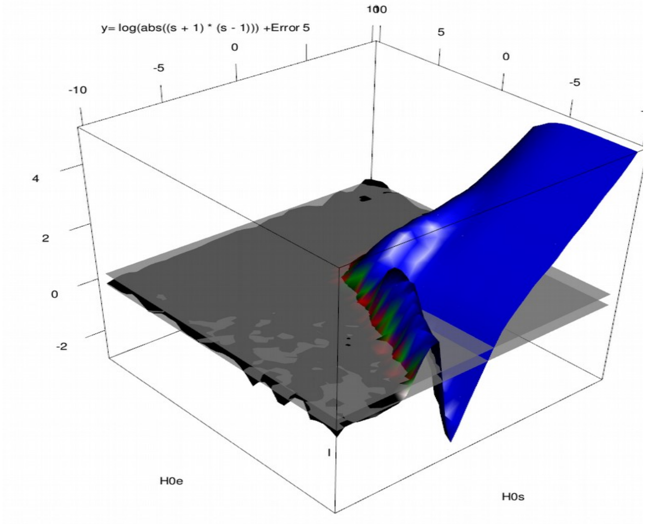

Hx+Hy-Hxy (y= sinh(s) +error)
H0e
H0x

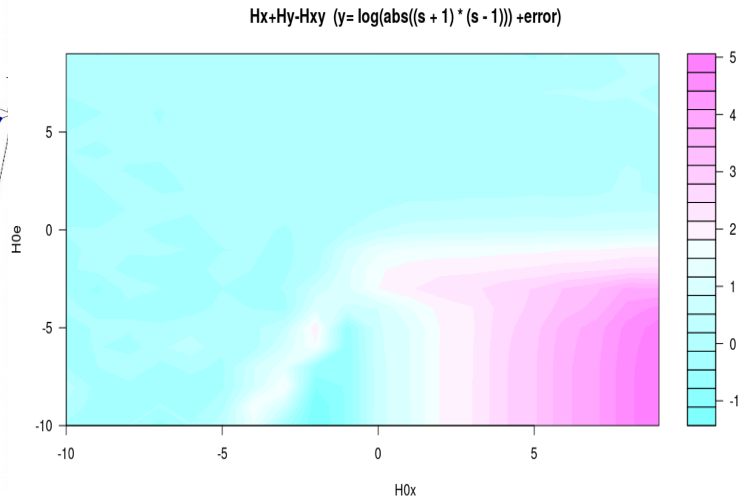

y= cosh(s) +Error
H0e
H0s

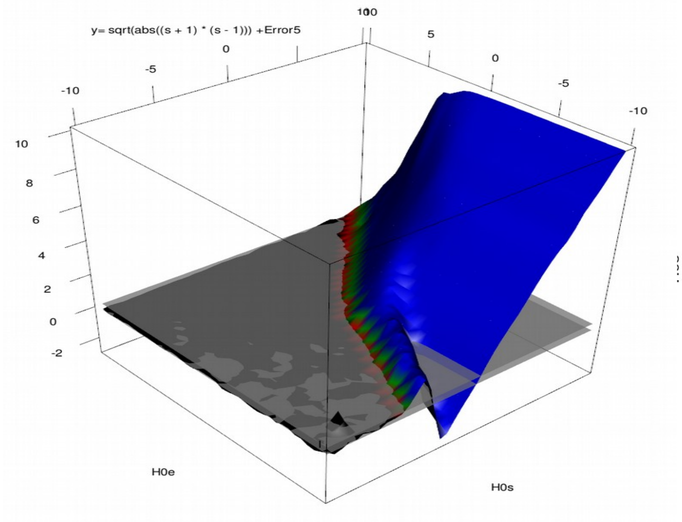

Hx+Hy-Hxy (y= cosh(s) +error)
H0x

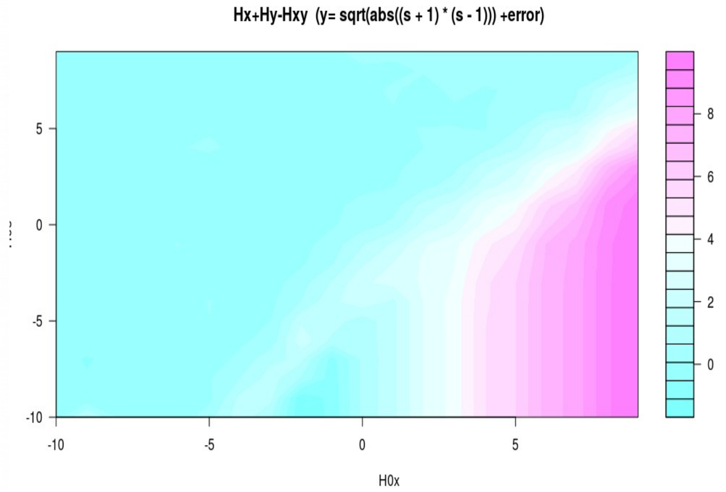

y= tanh(s) +Error
H0e
H0s

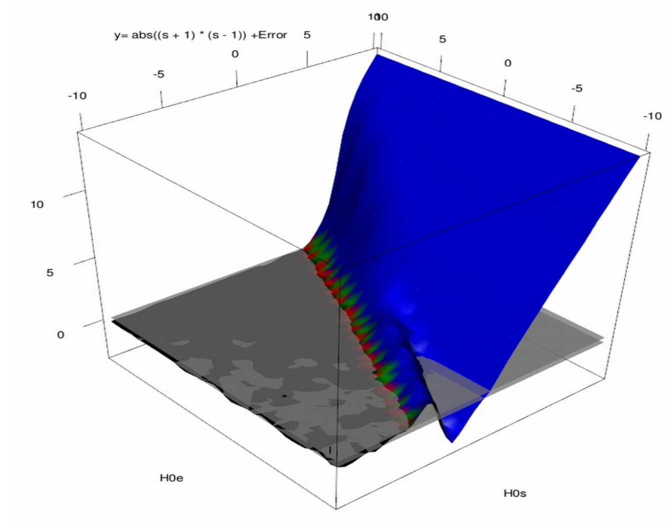

Hx+Hy-Hxy (y= tanh(s) +error)
H0e
H0x

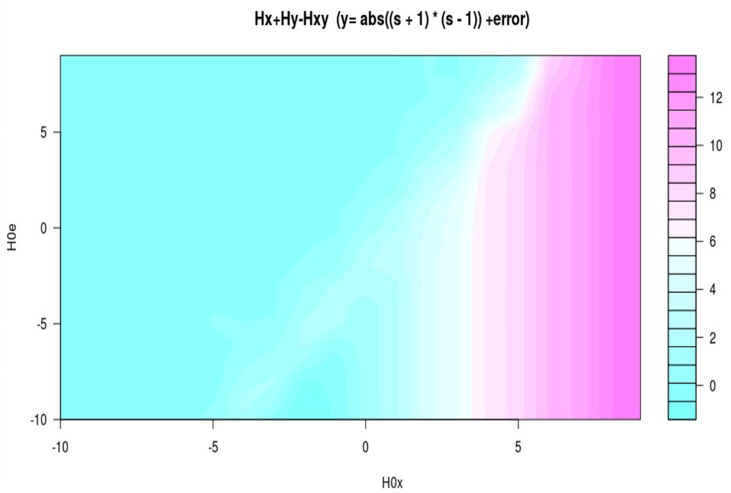

y= log(abs((s + 1) * (s - 1))) +Error
H0e
H0s

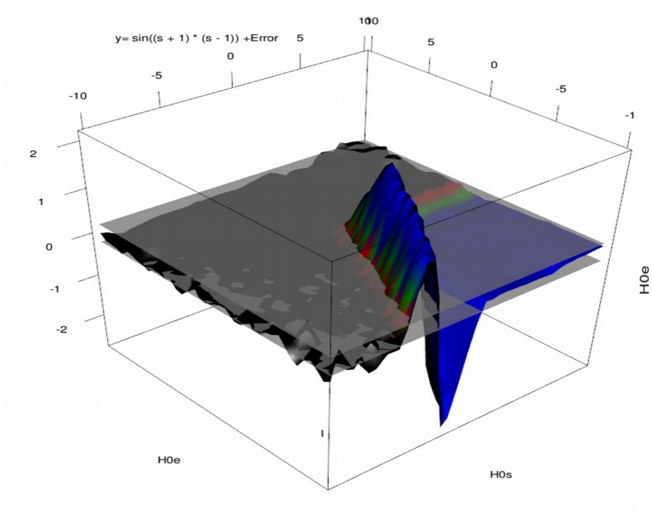

Hx+Hy-Hxy (y= log(abs((s + 1) * (s - 1))) +error)
H0e
H0x

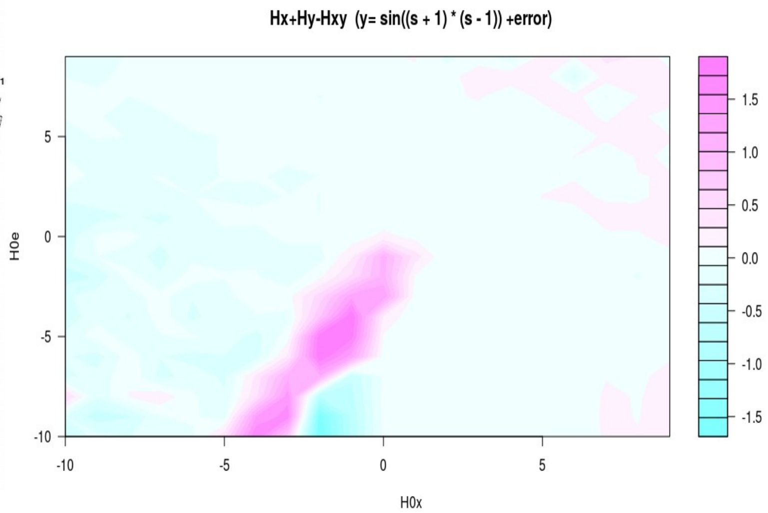

y= sqrt(abs((s + 1) * (s - 1))) +Error
H0e
H0s

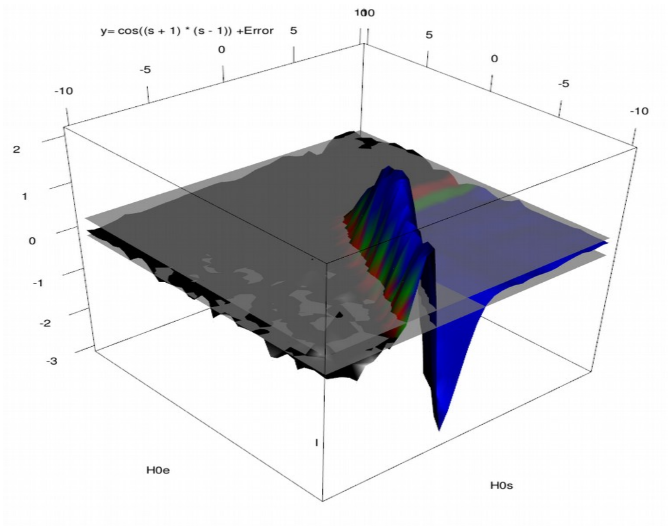

Hx+Hy-Hxy (y= sqrt(abs((s + 1) * (s - 1))) +error)
H0x

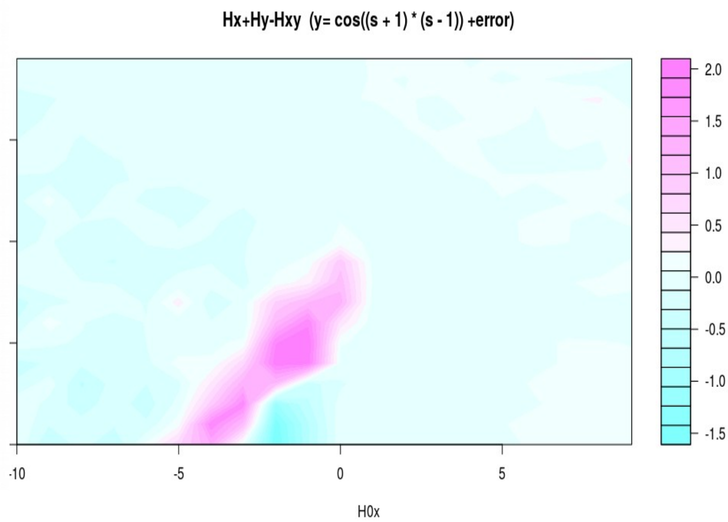

y= abs((s + 1) * (s - 1)) +Error
H0e
H0s

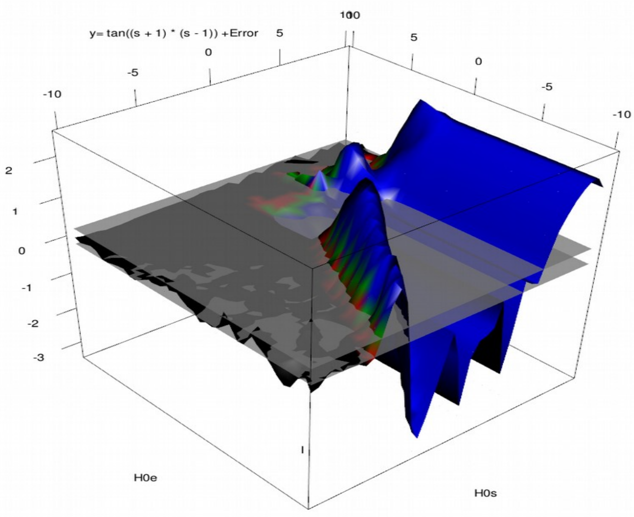

Hx+Hy-Hxy (y= abs((s + 1) * (s - 1)) +error)
H0e
H0x

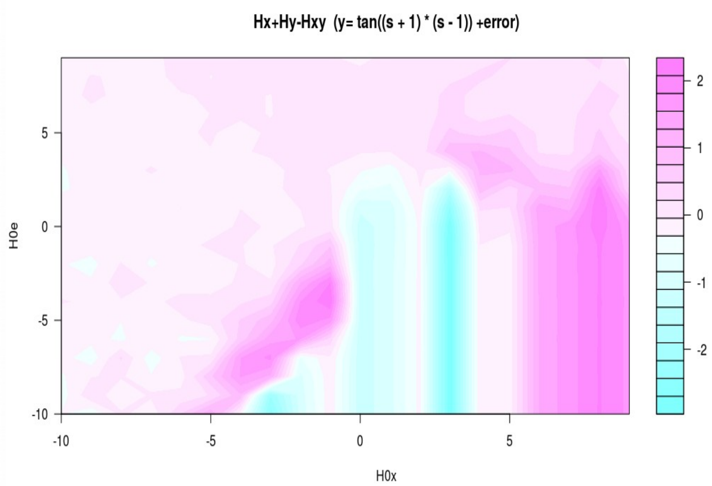

y= sin((s + 1) * (s - 1)) +Error
H0e
H0s

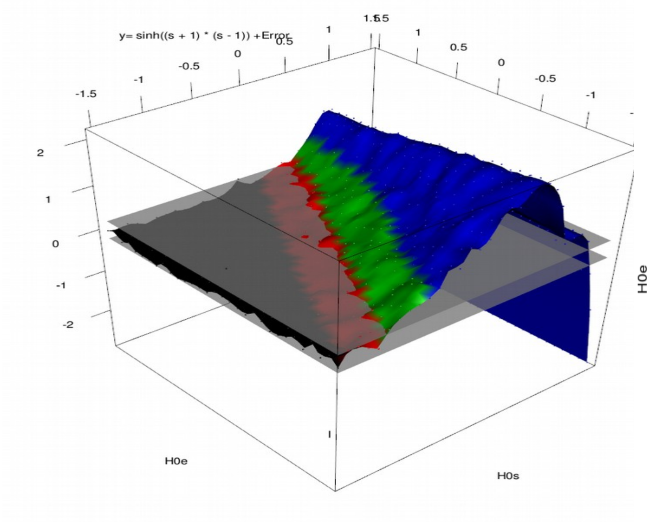

Hx+Hy-Hxy (y= sin((s + 1) * (s - 1)) +error)
H0e
H0x

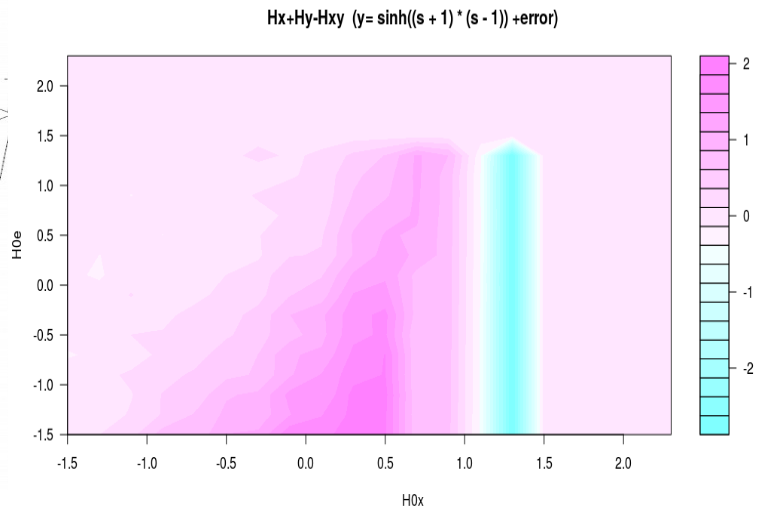

y= cos((s + 1) * (s - 1)) +Error
H0e
H0s

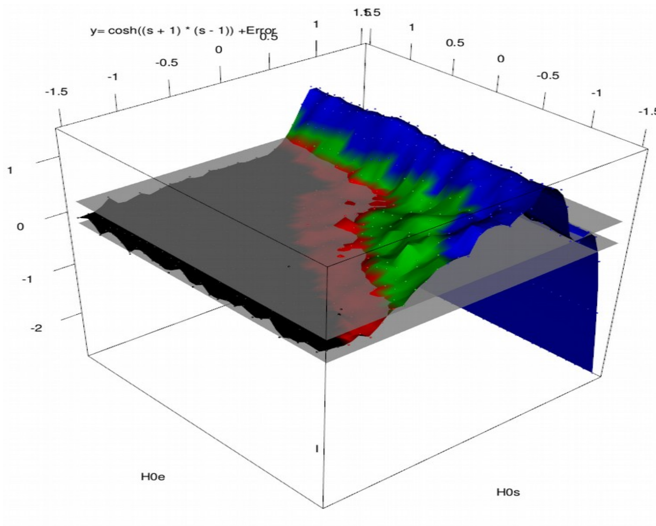

Hx+Hy-Hxy (y= cos((s + 1) * (s - 1)) +error)
H0x

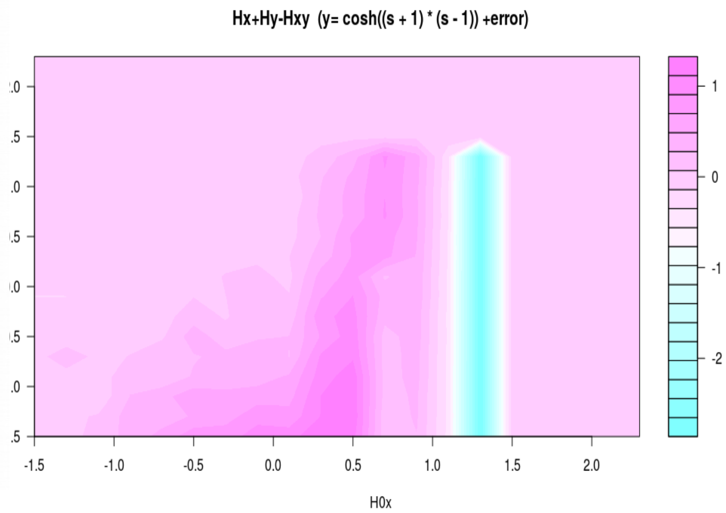

y= tan((s + 1) * (s - 1)) +Error
H0e
H0s

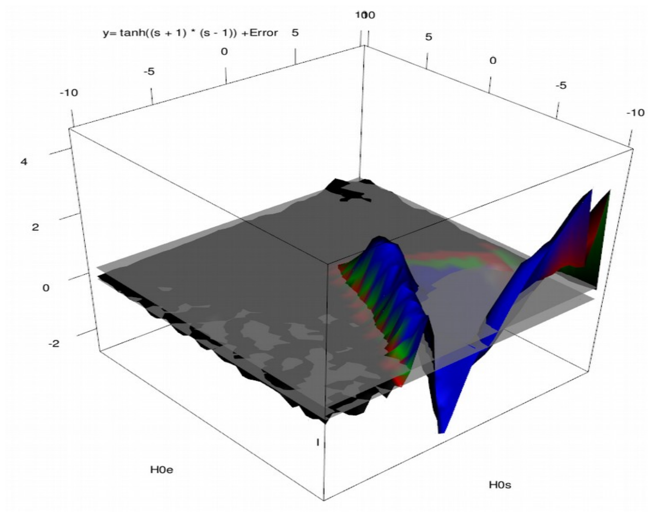

Hx+Hy-Hxy (y= tan((s + 1) * (s - 1)) +error)
H0e
H0x

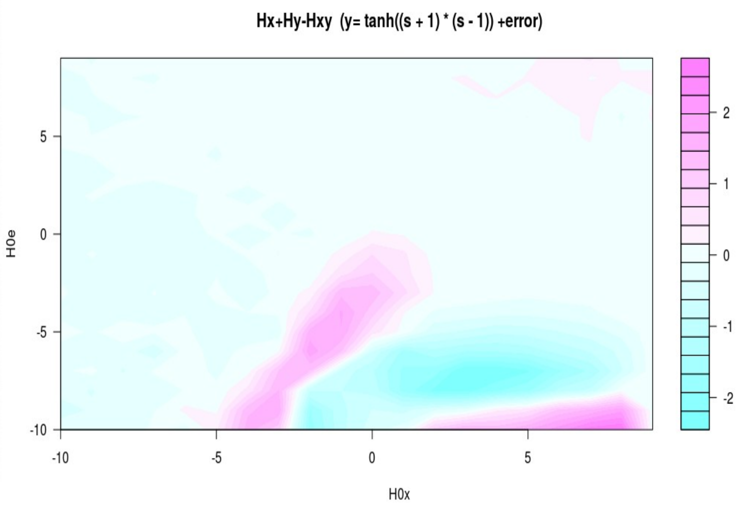

y= sinh((s + 1) * (s - 1)) +Error
H0e
H0s


Hx+Hy-Hxy (y= sinh((s + 1) * (s - 1)) +error)
H0e
H0x


y= cosh((s + 1) * (s - 1)) +Error
H0e
H0s


Hx+Hy-Hxy (y= cosh((s + 1) * (s - 1)) +error)
H0x


y= tanh((s + 1) * (s - 1)) +Error
H0e
H0s


Hx+Hy-Hxy (y= tanh((s + 1) * (s - 1)) +error)
H0e
H0x