# Image created for CHANCE

# Pixel by Pixel generative art contest

# <https://chance.amstat.org/2023/04/pixel-by-pixel>

# Giulio Stancari

# Senior Scientist

# Fermi National Accelerator Laboratory

# Batavia, Illinois, USA

# <stancari@fnal.gov>

# August 2023

library(viridis)

# simulation parameters

set.seed(3) # random seed

Nt = 128 # number of iterations

Np = 128 # number of particles

Npanels = 49 # number of panels (should be a perfect square)

alpha\_values = runif(Npanels, 0.24, 0.27) # set of rotation numbers

Knl = 0.7 # strength of the nonlinearity

Pnl = 3 # power of the nonlinearity

mi = -1; Ma = 1 # coordinate range

# graphical parameters

width\_in = 15 # width of the figure [in]

height\_in = 15 # height of the figure [in]

pointsize\_pt = 10 # point size for text

file\_basename = "tmp\_image\_composite" # file name

sy = 20 # symbol for points

sc = 0.01 # symbol scaling factor

colo = rev(rocket(Np)) # color palette

# set up plot

graphics.off()

dev.new(width=width\_in, height=height\_in)

par(oma=rep(0, 4), mar=rep(0, 4), bg="black")

layout(matrix(1:Npanels, nrow=sqrt(Npanels), ncol=sqrt(Npanels), byrow=TRUE))

for(alpha in alpha\_values){

# pre-calculated quantities

psi = 2\*pi\*alpha # phase advance

co = cos(psi)

si = sin(psi)

# initialize coordinates

X = Y = Amplitude = Phase = array(dim=c(Np, Nt))

# initial conditions

if(FALSE){

# choose amplitudes partly uniformly and partly emphasizing outer edges

uniform\_fraction = 1/4

a = runif(Np\*(1-uniform\_fraction))

Amplitude[, 1] = c( seq(0, Ma, length=Np\*uniform\_fraction), Ma \* a^0.05 )

# uniform phases

Phase[, ] = runif(Np, -pi, pi)

# derive coordinates and momenta

X[, 1] = Amplitude[, 1] \* cos(Phase[, 1])

Y[, 1] = Amplitude[, 1] \* sin(Phase[, 1])

} else {

# uniform horizontal and vertical distributions

X[, 1] = runif(Np, mi, Ma)

Y[, 1] = runif(Np, mi, Ma)

Amplitude[, 1] = sqrt(X[,1]^2+Y[,1]^2)

Phase[, 1] = atan2(Y[,1], X[,1])

}

# sort arrays by initial amplitude

ia = sort.int(Amplitude[, 1], index.return=TRUE)$ix

X[, 1] = X[ia, 1]

Y[, 1] = Y[ia, 1]

Amplitude[, 1] = Amplitude[ia, 1]

Phase[, 1] = Phase[ia, 1]

# iterate nonlinear map

for(i in 2:Nt){

X[, i] = X[, (i-1)]\*co + Y[, (i-1)]\*si

Y[, i] = -X[, (i-1)]\*si + Y[, (i-1)]\*co + Knl\*X[, i]^Pnl

}

Amplitude = sqrt(X^2+Y^2)

Phase = atan2(Y, X)

# plot coordinates

plot(X, Y,

xlim=c(mi, Ma), ylim=c(mi, Ma),

axes=FALSE, asp=1, xlab="", ylab="",

pch=sy, cex=sc, col=colo, xpd=NA)

}

dev2bitmap(file=paste0(file\_basename, ".png"), type='png16m',

width=width\_in, height=height\_in,

pointsize=pointsize\_pt, res=300)

dev.off()