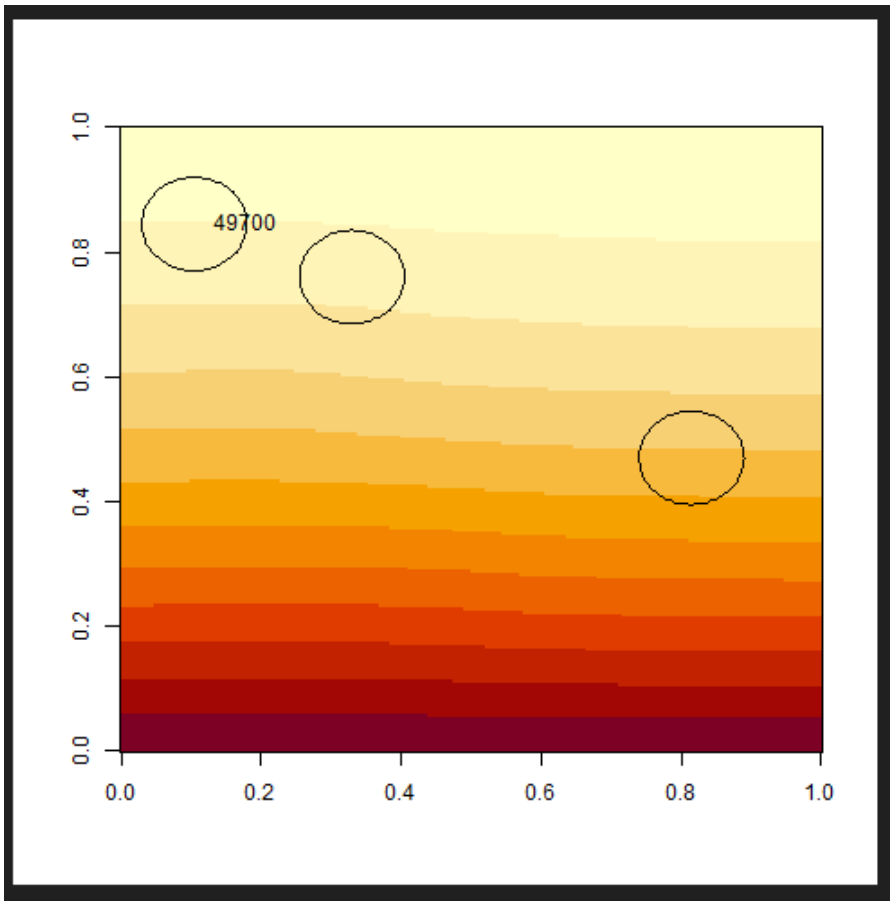
Sprawozdanie - model geotermalny.

Radosław Beta, Teresa Holerek

3 kwietnia 2024

1 Efekt

W wyniku naszej pracy otrzymaliśmy wynik końcowy widoczny na zdjęciu. Oprócz sprawozdania wysyłamy mailem również plik całej animacji.



Rysunek 1: Jest to ujęcie prawie ustabilizowanego ośrodka w trakcie końcowych iteracji.

2 Kod

W tej sekcji opiszemy poszczególne elementy kodu napisanego w języku R.

2.1 Ustalenie wymiaru, kroku i rozpoczęcie mierzenia czasu.

N<-200
h<-5
image(t(apply(a, 2, rev)))
start<-Sys.time()

2.2 Określenie parametrów okręgów.

x <- sample(26:174, 6)
a<-circle-matrix(N, N, c(x[1],x[2],x[3]), c(x[4],x[5],x[6]), c(25,25,25), c(6,6,6))
a[a == 0] <- 2

2.3 Inicjalizacja macierzy dla kroku nowego i przeszłego.

L<-matrix(nrow=N,ncol=N,2)
Lnew<-matrix(nrow=N,ncol=N,2)

2.4 Warunki brzegowe.

L[,1]<-rep(0,N)
L[,N]<-rep(0,N)
L[1,]<-rep(0,N)
L[N,]<-rep(50,N)

2.5 Max a do stabilności.

a-max<-max(a)

2.6 Krok czasowy.

dt<-h^2/(4*a-max)

2.7 Czas symulacji.

t<-0

2.8 Ustalenie warunków brzegowych również w nowym kroku.

Lnew <- L

2.9 Liczba iteracji.

iter<-25000

2.10 Inicjalizacja paska postępu.

prog-bar<-txtProgressBar(min=0,max=iter,style=3)
image(t(apply(a, 2, rev)))

2.11 Określenie promieni okręgów i ich liczby.

num-of-circles <- 3
r <- 15

2.12 Stworzenie list do przechowywania współrzędnych kółek.

x-list <- list()
y-list <- list()
for (k in 1:num-of-circles)
x1 <- sample((r + 1):(N - r), 1)
y1 <- sample((r + 1):(N - r), 1)

2.12.1 Skalowanie

x2 <- x1 / N
y2 <- y1 / N
for (i in (x1 - r):(x1 + r))
for (j in (y1 - r):(y1 + r))
if $((i - x1)^2 + (j - y1)^2 <= r^2)$
a[i, j] <- 6

2.12.2 Zapisanie współrzędnych okęgu.

x-list[[k]] <- 1 - x2
y-list[[k]] <- y2

2.13 Tworzenie GIFa.

image(t(apply(a, 2, rev)))
saveGIF(
step<-(-1)
for (k in 1:iter)
t<-t+dt
step<-step+1
setTxtProgressBar(prog-bar, step)

2.13.1 Pętla po wierszach i kolumnach.

for (i in 2:(N-1))
for (j in 2:(N-1))
Lnew[i,j]<-$(1-(4*dt*a[i,j])/(h^2))*L[i,j]+dt*a[i,j]*((L[i-1,j]+L[i+1,j]+L[i,j-1]+L[i,j+1])/(h^2))$

2.13.2 Gradient.

Lnew[,1]<-L[,2]
Lnew[,N]<-L[,N-1]

2.13.3 PreL zachowana macierz do badania stabilności, przejście o krok w iteracji.

preL<-L
L<-Lnew

2.13.4 Rotacja obrazu na pionowy.

if (k dzielone 100==0)
Limg <- apply(L, 2, rev)
image(t(Limg))

2.13.5 Dodanie w lewym górnym rogu ilości iteraji.

text(0.18,0.85,k)

2.13.6 Wyrysowanie kółek.

for (i in 1:num-of-circles)
circle <- seq(0, 2 * pi, length.out = 100)
x <- (y-list[[i]] + r / N * sin(circle))
y <- (x-list[[i]] + r / N * cos(circle))

polygon(x, y)

2.13.7 Czarne ramki.

box()
box()

2.13.8 Dodanie parametru interwału do funkcji saveGIF.

,interval=0.1)

2.13.9 Zakończenie kodu.

stop<-Sys.time()