Algorithm for cartogram displacement field construction

Constants and variables

▷ Length of map in x and y directions
integer LX,LY

▷ Array for densities
float rho_0[$LX + 1$][$LY + 1$], rho[$LX + 1$][$LY + 1$]
▷ rho_0[j][k] is the density at $t = 0$, rho[j][k] that at $t > 0$.

▷ Arrays for the velocity field at integer-valued positions
float gridvx[$LX + 1$][$LY + 1$], gridvy[$LX + 1$][$LY + 1$]
▷ gridvx[j][k] is the velocity component in x direction at position (j, k). Similarly for y.

▷ Arrays for the position at $t > 0$
float x[$LX + 1$][$LY + 1$], y[$LX + 1$][$LY + 1$]
▷ x[j][k] is the x-coordinate for the element that was at position (j, k) at time $t = 0$.

▷ Arrays for the velocity field at position (x[j][k], y[j][k])
float vx[$LX + 1$][$LY + 1$], vx[$LX + 1$][$LY + 1$]
▷ vx[j][k] is the velocity component in x direction at the position (x[j][k], y[j][k]).

Program CARTOGRAM

Initialize rho_0.
If wanted, perform Gaussian blur by Fast Fourier Transform.
Replace rho_0 by cosine Fourier transform in both variables.
$t \leftarrow 0$ ▷ Initialize time.
$h \leftarrow HINITIAL$ ▷ Initialize time step size.
▷ Initialize x, y, vx and vy.
for $j \leftarrow 0$ **to** LX
  do for $k \leftarrow 0$ **to** LY
    do x[j][k] $\leftarrow j$
      y[j][k] $\leftarrow k$
Call subroutine CALCV(time = 0.0).
for $j \leftarrow 0$ **to** LX
  do for $k \leftarrow 0$ **to** LY
    do vx[j][k] $\leftarrow$ gridvx[j][k]
      vy[j][k] $\leftarrow$ gridvy[j][k]

while the position arrays x and y have not sufficiently converged
do Call subroutine CALCV(time $= t + h$).
for $j \leftarrow 0$ **to** LX
do for $k \leftarrow 0$ **to** LY
do $\triangleright$ Find the new positions in the following manner. First we take a naive integration step:
$vxminus \leftarrow$ vx[j][k]
$vyminus \leftarrow$ vy[j][k]
$vxplus \leftarrow \vec{v}_x(t + h,$ x[j][k]+h*vx[j][k], y[j][k]+h*vy[j][k])
$vyplus \leftarrow \vec{v}_y(t + h,$ x[j][k]+h*vx[j][k], y[j][k]+h*vy[j][k]),
$\triangleright$ where the velocity $\vec{v}$ at time $t + h$ and position (x[j][k]+h*vx[j][k], y[j][k]+ h*vy[j][k]) can be interpolated from the arrays gridvx and gridvy. Then we expect the new position at time $t + h$ to be:
$xguess \leftarrow$ x[j][k] + 0.5*h*($vxminus$+$vxplus$)
$yguess \leftarrow$ y[j][k] + 0.5*h*($vyminus$+$vyplus$)
$\triangleright$ Then we make a better approximation by solving the two nonlinear equations:
xappr[j][k] - 0.5*h*$\vec{v}_x(t + h,$ xappr[j][k], yappr[j][k]) - x[j][k] - 0.5*h*vx[j][k] $= 0$,
yappr[j][k] - 0.5*h*$\vec{v}_y(t + h,$ xappr[j][k], yappr[j][k]) - y[j][k] - 0.5*h*vy[j][k] $= 0$
$\triangleright$ simultaneously for xappr[j][k], yappr[j][k], e.g., using the Newton-Raphson method with ($xguess$, $yguess$) as initial guess. The velocity $\vec{v}$ at time $t + h$ and position (xappr[j][k],yappr[j][k]) can again be interpolated from the arrays gridvx and gridvy. If ($xguess$, $yguess$) and (xappr[j][k], yappr[j][k]) differ by more than some predefined tolerance, reduce step size h, break, and try again.
$t \leftarrow t + h$
for $j \leftarrow 0$ **to** LX
do for $k \leftarrow 0$ **to** LY
do x[j][k] $\leftarrow$ xappr[j][k]
y[j][k] $\leftarrow$ yappr[j][k]
vx[j][k] $\leftarrow \vec{v}_x(t + h,$ xappr[j][k], yappr[j][k])
vy[j][k] $\leftarrow \vec{v}_y(t + h,$ xappr[j][k], yappr[j][k])
Increase step size h.

Subroutine CALCV(time t)

▷ First calculate the density rho by filling the array with the Fourier coefficients.

for $j \leftarrow 0$ **to** LX

  do for $k \leftarrow 0$ **to** LY

    do rho[j][k] $\leftarrow$

      exp(-(($\pi * j/LX$)*($\pi * j/LX$)+($\pi * k/LY$)*($\pi * k/LY$))*t)*rho_0[j][k]

▷ Calculate the Fourier coefficients for the partial derivative of rho. Store temporary result in arrays gridvx, gridvy.

for $j \leftarrow 0$ **to** LX

  do for $k \leftarrow 0$ **to** LY

    do gridvx[j][k] $\leftarrow$ -($\pi * j/LX$)*rho[j][k]

     gridvy[j][k] $\leftarrow$ -($\pi * k/LY$)*rho[j][k]

Replace rho by cosine Fourier backtransform in both variables.

Replace vx by sine Fourier backtransform in the first and cosine Fourier backtransform in the second variable.

Replace vy by cosine Fourier backtransform in the first and sine Fourier backtransform in the second variable.

▷ Calculate the velocity field.

for $j \leftarrow 0$ **to** LX

  do for $k \leftarrow 0$ **to** LY

    do gridvx[j][k] $\leftarrow$ -gridvx[j][k]/rho[j][k]

     gridvy[j][k] $\leftarrow$ -gridvy[j][k]/rho[j][k]