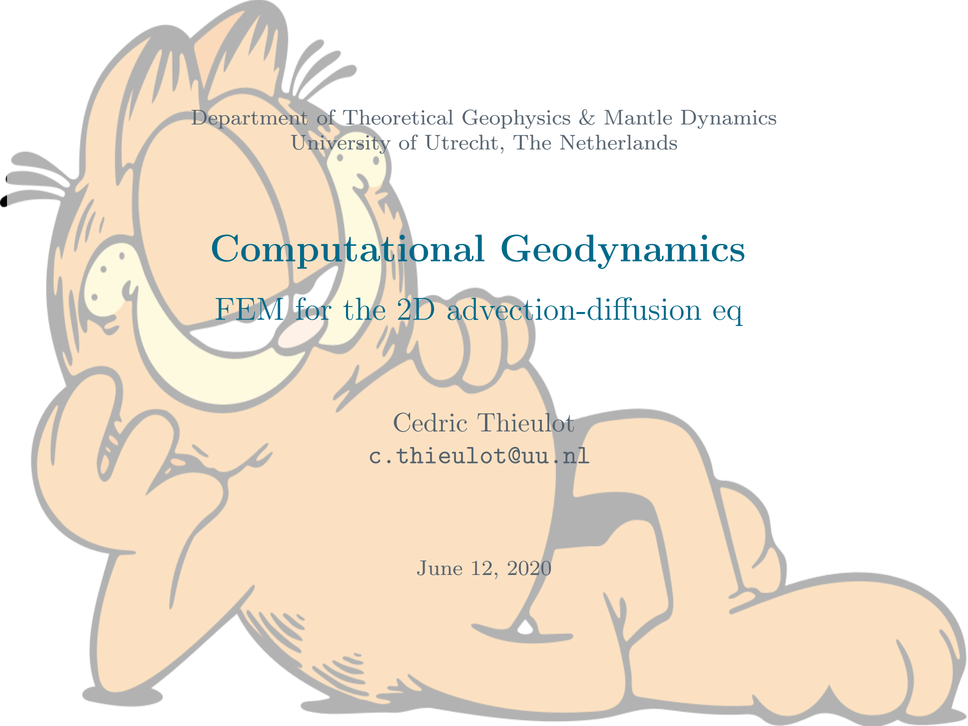


Department of Theoretical Geophysics & Mantle Dynamics
University of Utrecht, The Netherlands

Computational Geodynamics

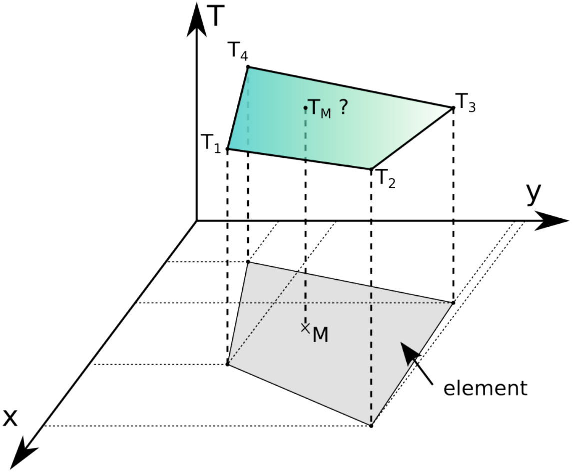
FEM for the 2D advection-diffusion eq

Cedric Thieulot
c.thieulot@uu.nl

June 12, 2020

Content

FEM in 2D



What is the temperature at a point $M(x, y)$ inside the element ?

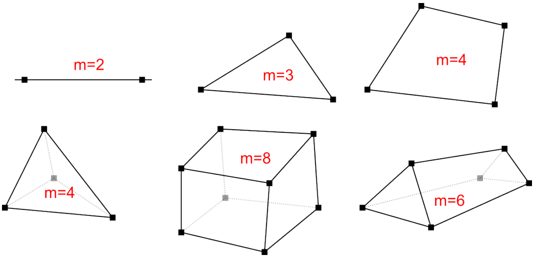
FEM in 2D

The temperature inside an element is given by:

$$T(\vec{r}) = \sum_{k=1}^{m} N_k(\vec{r}) T_k = \vec{N} \cdot \vec{T}$$

where

$$\vec{N} = (N_1(\vec{r}) \; N_2(\vec{r}) \; N_3(\vec{r}) \; \dots \; N_m(\vec{r})) \qquad \vec{T}^T = (T_1 \quad T_2 \quad T_3 \quad \dots \quad T_m)$$



FEM in 2D

From the vector of shape functions,

$$\vec{N} = \begin{pmatrix} N_1 & N_2 & N_3 & \dots & N_m \end{pmatrix}$$

one can build the gradient matrix (in Cartesian coords.):

$$\boldsymbol{B} = \vec{\nabla}\vec{N} = \begin{pmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} & \dots & \frac{\partial N_m}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} & \dots & \frac{\partial N_m}{\partial y} \\ \frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial z} & \frac{\partial N_3}{\partial z} & \dots & \frac{\partial N_m}{\partial z} \end{pmatrix}$$

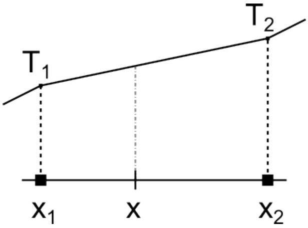
Its size is $ndim \times m$

in 1D, for $x_1 \leq x \leq x_2$

$$T(x) = N_1(x) T_1 + N_2(x) T_2$$

with

$$N_1(x) = \frac{x_2 - x}{h_x} \qquad N_2(x) = \frac{x - x_1}{h_x}$$



in 2D, in a quadrilateral:

$$T(x,y) = N_1(x,y)T_1 + N_2(x,y)T_2 + N_3(x,y)T_3 + N_4(x,y)T_4$$

with

$$\begin{aligned}
N_1(x,y) &= \left(\frac{x_3 - x}{h_x}\right)\left(\frac{y_3 - y}{h_y}\right) \\
N_2(x,y) &= \left(\frac{x - x_1}{h_x}\right)\left(\frac{y_3 - y}{h_y}\right) \\
N_3(x,y) &= \left(\frac{x - x_1}{h_x}\right)\left(\frac{y - y_1}{h_y}\right) \\
N_4(x,y) &= \left(\frac{x_3 - x}{h_x}\right)\left(\frac{y - y_1}{h_y}\right)
\end{aligned}$$

in 2D, in a quadrilateral:

$$T(x,y) = N_1(x,y)T_1 + N_2(x,y)T_2 + N_3(x,y)T_3 + N_4(x,y)T_4$$

with

$$\begin{aligned}
N_1(x,y) &= \left(\frac{x_3 - x}{h_x}\right)\left(\frac{y_3 - y}{h_y}\right) \\
N_2(x,y) &= \left(\frac{x - x_1}{h_x}\right)\left(\frac{y_3 - y}{h_y}\right) \\
N_3(x,y) &= \left(\frac{x - x_1}{h_x}\right)\left(\frac{y - y_1}{h_y}\right) \\
N_4(x,y) &= \left(\frac{x_3 - x}{h_x}\right)\left(\frac{y - y_1}{h_y}\right)
\end{aligned}$$

At point 1, of coordinates (x_1, y_1), we have

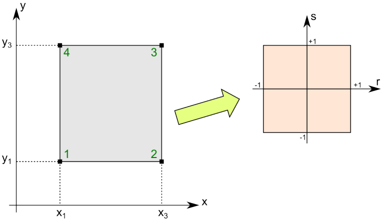
$$N_1(x_1,y_1) = 1 \qquad N_2(x_1,y_1) = 0 \qquad N_3(x_1,y_1) = 0 \qquad N_4(x_1,y_1) = 0$$

FEM in 2D

we can then compute the gradient matrix $\boldsymbol{B}$ in 2D:

$$
\boldsymbol{B}(x,y) = \begin{pmatrix}
\frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial x} \\
\frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} & \frac{\partial N_4}{\partial y}
\end{pmatrix}
= \begin{pmatrix}
-\frac{1}{h_x}\frac{y_3-y}{h_y} & \frac{1}{h_x}\frac{y_3-y}{h_y} & \frac{1}{h_x}\frac{y-y_1}{h_y} & -\frac{1}{h_x}\frac{y-y_1}{h_y} \\
-\frac{1}{h_y}\frac{x_3-x}{h_x} & -\frac{1}{h_y}\frac{x-x_1}{h_x} & \frac{1}{h_y}\frac{x-x_1}{h_x} & \frac{1}{h_y}\frac{x_3-x}{h_x}
\end{pmatrix}
$$

FEM in 2D



Carrying out a change of variables:

$$x \in [x_1 : x_3] \quad \rightarrow \quad r \in [-1 : 1]$$

$$y \in [y_1 : y_3] \quad \rightarrow \quad s \in [-1 : 1]$$

so that

$$r = \frac{2}{h_x}(x - x_1) - 1 \qquad\qquad s = \frac{2}{h_y}(y - y_1) - 1$$

FEM in 2D

Then

$$
\begin{aligned}
N_1(r,s) &= \frac{1}{4}(1-r)(1-s)\\
N_2(r,s) &= \frac{1}{4}(1+r)(1-s)\\
N_3(r,s) &= \frac{1}{4}(1+r)(1+s)\\
N_4(r,s) &= \frac{1}{4}(1-r)(1+s)
\end{aligned}
$$

and

$$
\boldsymbol{B}(r,s) = \begin{pmatrix}
-\frac{1}{h_x}\frac{1}{2}(1-s) & \frac{1}{h_x}\frac{1}{2}(1-s) & \frac{1}{h_x}\frac{1}{2}(1+s) & -\frac{1}{h_x}\frac{1}{2}(1+s) \\
-\frac{1}{h_y}\frac{1}{2}(1-r) & -\frac{1}{h_y}\frac{1}{2}(1+r) & \frac{1}{h_y}\frac{1}{2}(1+r) & \frac{1}{h_y}\frac{1}{2}(1-r)
\end{pmatrix}
$$

FEM in 2D

Finally

$$\int_{\Omega_e} \dots d\Omega = \int_{x_1}^{x_3} \int_{y_1}^{y_3} \dots dxdy$$
$$= \frac{h_x h_y}{4} \int_{-1}^{+1} \int_{-1}^{+1} \dots drds$$

FEM in 2D

The strong form of the heat transport equation in 1,2,3 dimensions writes:

$$\rho c_p \left(\frac{\partial T}{\partial t} + \vec{v} \cdot \vec{\nabla} T \right) = \vec{\nabla} \cdot (k \vec{\nabla} T) + H$$

FEM in 2D

The strong form of the heat transport equation in 1,2,3 dimensions writes:

$$\rho c_p \left(\frac{\partial T}{\partial t} + \vec{v} \cdot \vec{\nabla} T \right) = \vec{\nabla} \cdot (k \vec{\nabla} T) + H$$

The weak form writes:

$$\int_\Omega \rho c_p f(\vec{r}) \dot{T} d\Omega + \int_\Omega \rho c_p f(\vec{r}) \vec{v} \cdot \vec{\nabla} T d\Omega = \int_\Omega f(\vec{r}) \vec{\nabla} \cdot (k \vec{\nabla} T) d\Omega + \int_\Omega f(\vec{r}) H d\Omega$$

FEM in 2D

The strong form of the heat transport equation in 1,2,3 dimensions writes:

$$\rho c_p \left(\frac{\partial T}{\partial t} + \vec{v} \cdot \vec{\nabla} T \right) = \vec{\nabla} \cdot (k \vec{\nabla} T) + H$$

The weak form writes:

$$\int_\Omega \rho c_p f(\vec{r}) \dot{T} d\Omega + \int_\Omega \rho c_p f(\vec{r}) \vec{v} \cdot \vec{\nabla} T d\Omega = \int_\Omega f(\vec{r}) \vec{\nabla} \cdot (k \vec{\nabla} T) d\Omega + \int_\Omega f(\vec{r}) H d\Omega$$

We integrate by parts the diffusion term and neglect here again the surface term so that:

$$\int_\Omega \rho c_p f(\vec{r}) \dot{T} d\Omega + \int_\Omega \rho c_p f(\vec{r}) \vec{v} \cdot \vec{\nabla} T d\Omega + \int_\Omega \vec{\nabla} f(\vec{r}) \cdot (k \vec{\nabla} T) d\Omega = \int_\Omega f(\vec{r}) H d\Omega$$

We then use the additive property of the integral:

$$\int_\Omega \cdots = \sum_{elts} \int_{\Omega_e} \cdots$$

so that

$$\underbrace{\int_{\Omega_e} \rho c_p f(\vec{r}) \dot{T} d\Omega}_{\Lambda_f^e} + \underbrace{\int_\Omega \rho c_p f(\vec{r}) \vec{v} \cdot \vec{\nabla} T d\Omega}_{\Sigma_f^e} + \underbrace{\int_\Omega \vec{\nabla} f(\vec{r}) \cdot (k \vec{\nabla} T) d\Omega}_{\Upsilon_f^e} = \underbrace{\int_\Omega f(\vec{r}) H d\Omega}_{\Omega_f^e}$$

FEM in 2D

Let us then compute separately

$$\Lambda_f^e = \int_{\Omega_e} \rho c_p f(\vec{r}) \dot{T} d\Omega$$

$$\Sigma_f^e = \int_{\Omega} \rho c_p f(\vec{r}) \vec{v} \cdot \vec{\nabla} T d\Omega$$

$$\Upsilon_f^e = \int_{\Omega} \vec{\nabla} f(\vec{r}) \cdot (k \vec{\nabla} T) d\Omega$$

$$\Omega_f^e = \int_{\Omega} f(\vec{r}) H(x,y) d\Omega$$

FEM in 2D

$$
\begin{aligned}
\Lambda_f^e &= \int_{\Omega_e} \rho c_p f(\vec{r}) \dot{T} d\Omega \\
&= \int_{\Omega_e} \rho c_p f(\vec{r}) \sum_{k=1}^{m} N_k(\vec{r}) \dot{T}_k d\Omega \\
&= \left(\int_{\Omega_e} \rho c_p f(\vec{r}) \vec{N} d\Omega \right) \cdot \dot{\vec{T}}
\end{aligned}
$$

Letting $f(\vec{r}) \rightarrow \vec{N}^T$ then

$$
\Lambda^e = \left(\int_{\Omega_e} \rho c_p \vec{N}^T \vec{N} d\Omega \right) \cdot \dot{\vec{T}}
$$

$\vec{N}$ has size m, so $\vec{N}^T \vec{N}$ is a matrix of size $m \times m$.

FEM in 2D

$$
\begin{aligned}
\Sigma_f^e &= \int_\Omega \rho c_p \boldsymbol{f}(\vec{r})\vec{v}\cdot\vec{\nabla} T d\Omega \\
&= \int_\Omega \rho c_p \boldsymbol{f}(\vec{r})\vec{v}\cdot\vec{\nabla}(\sum_k N_k T_k) d\Omega \\
&= \sum_k \int_\Omega \rho c_p \boldsymbol{f}(\vec{r})\vec{v}\cdot\vec{\nabla} N_k \; T_k d\Omega \\
&= \left(\int_\Omega \rho c_p \boldsymbol{f}(\vec{r})\vec{v}\cdot\vec{\nabla}\vec{N} d\Omega \right)\cdot\vec{T} \\
&= \left(\int_\Omega \rho c_p \boldsymbol{f}(\vec{r})\vec{v}\cdot \boldsymbol{B} d\Omega \right)\cdot\vec{T}
\end{aligned}
$$

Letting $\boldsymbol{f}(\vec{r}) \rightarrow \vec{N}^T$ then

$$
\Sigma^e = \left(\int_\Omega \rho c_p \vec{N}^T (\vec{v}\cdot\boldsymbol{B}) d\Omega \right)\cdot\vec{T}
$$

FEM in 2D

$$
\begin{aligned}
\Upsilon_f^e &= \int_{\Omega_e} \vec\nabla f(\vec r)\cdot(k\vec\nabla T)d\Omega \\
&= \int_{\Omega_e} \vec\nabla f(\vec r)\cdot(k\vec\nabla(\sum_k N_k(\vec r)T_k))d\Omega \\
&= \sum_k \int_{\Omega_e} \vec\nabla f(\vec r)\cdot(k\vec\nabla N_k(\vec r))T_k d\Omega \\
&= \left(\int_{\Omega_e} \vec\nabla f(\vec r)\cdot(k\vec\nabla \vec N)d\Omega\right)\cdot\vec T \\
&= \left(\int_{\Omega_e} \vec\nabla f(\vec r)\cdot(k\boldsymbol{B})d\Omega\right)\cdot\vec T
\end{aligned}
$$

Letting $f(\vec r)\rightarrow \vec N^T$ then

$$
\Upsilon^e = \left(\int_{\Omega_e} k\boldsymbol{B}^T\cdot\boldsymbol{B}d\Omega\right)\cdot\vec T
$$

FEM in 2D

$$\Omega_f^e = \int_{\Omega_e} f(\vec{r}) H(x,y) d\Omega$$

Letting $f(\vec{r}) \rightarrow \vec{N}^T$ then

$$\Omega^e = \int_{\Omega_e} \vec{N}^T H(x,y) d\Omega$$

FEM in 2D

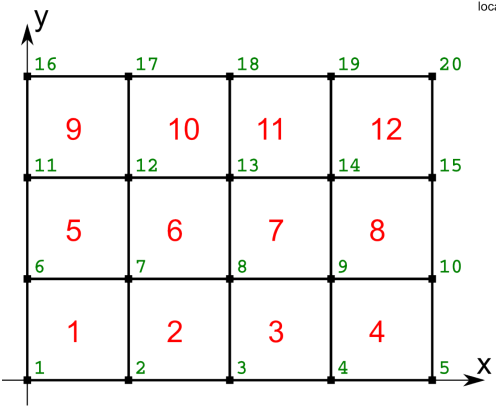
The weak form then writes:

$$\left(\int_{\Omega_e} \rho c_p \vec{N}^T \vec{N} d\Omega\right) \cdot \dot{\vec{T}} + \left(\int_{\Omega} \rho c_p \vec{N}^T \vec{v} \cdot \boldsymbol{B} d\Omega\right) \cdot \vec{T}$$

$$+ \left(\int_{\Omega_e} k \boldsymbol{B}^T \cdot \boldsymbol{B} d\Omega\right) \cdot \vec{T} = \int_{\Omega_e} \vec{N}^T H(x,y) d\Omega$$

or,

$$\boldsymbol{M}^e \cdot \dot{\vec{T}}^e + (\boldsymbol{K}_d^e + \boldsymbol{K}_a^e) \cdot \vec{T}^e = \vec{F}^e$$



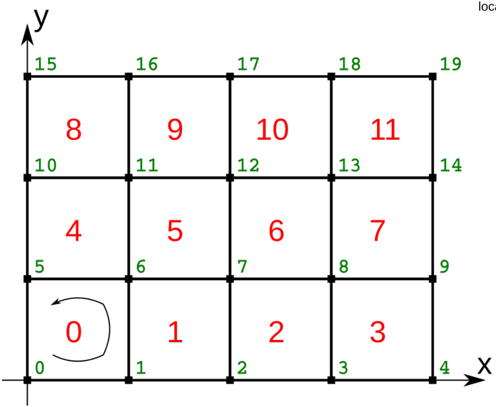
(Fortran)

local numbering
global numbering

icon(1,1)=1
icon(1,2)=2
icon(1,3)=7
icon(1,4)=6

...

icon(7,1)=8
icon(7,2)=9
icon(7,3)=14
icon(7,4)=13



local numbering
global numbering

icon[0,0]=0
icon[0,1]=1
icon[0,2]=6
icon[0,3]=5

...

icon[6,0]=7
icon[6,1]=8
icon[6,2]=13
icon[6,3]=12

(Python)

FEM in 2D

.

Since there are four vertices per element ($m = 4$), then $\vec{N}$ is of length 4 and $\boldsymbol{M}^e$ is a 4x4 matrix.

$$\begin{aligned}
\boldsymbol{M}^e &= \int_{\Omega_e} \rho c_p \vec{N}^T \vec{N} d\Omega \\
&= \int_{x_1}^{x_3} \int_{y_1}^{y_3} \rho c_p \vec{N}^T(x,y) \vec{N}(x,y) dxdy \\
&= \rho c_p \frac{h_x h_y}{4} \int_{-1}^{+1} \int_{-1}^{+1} \vec{N}^T(r,s) \vec{N}(r,s) drds
\end{aligned}$$

FEM in 2D

so

$$
\begin{aligned}
M_{11}^e &= \rho c_p \frac{h_x h_y}{4} \int_{-1}^{+1} \int_{-1}^{+1} N_1(r,s) N_1(r,s) dr ds \\
&= \rho c_p \frac{h_x h_y}{4} \int_{-1}^{+1} \int_{-1}^{+1} \frac{1}{16} (1-r)^2 (1-s)^2 dr ds \\
&= \rho c_p \frac{h_x h_y}{64} \int_{-1}^{+1} \int_{-1}^{+1} (1-r)^2 (1-s)^2 dr ds \\
&= \rho c_p \frac{h_x h_y}{64} \int_{-1}^{+1} (1-r)^2 dr \int_{-1}^{+1} (1-s)^2 ds \\
&= \rho c_p \frac{h_x h_y}{64} \frac{8}{3} \frac{8}{3} \\
&= \rho c_p \frac{h_x}{3} \frac{h_y}{3}
\end{aligned}
$$

Likewise we arrive at:

$$\boldsymbol{M}^e = \rho c_p \frac{h_x}{3}\frac{h_y}{3} \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix}$$

FEM in 2D

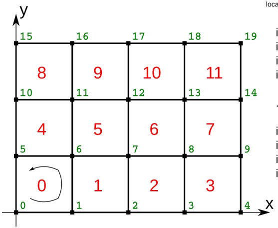
$$\boldsymbol{K}_d^e = \int_{\Omega_e} k \boldsymbol{B}^T \cdot \boldsymbol{B} d\Omega$$

$\boldsymbol{B}$ is a 2x4 matrix ($ndim \times m$) so that $\boldsymbol{B}^T \cdot \boldsymbol{B}$ is a 4x4 matrix.

FEM in 2D
Exercise 1

Compute the analytical values of $\boldsymbol{M}$, $\boldsymbol{K}_a$ and $\boldsymbol{K}_d$ for rectangular elements of size h_x, h_y.

Set nelx=4 and nely=3 (for now) and write a double for loop which automatically fills the icon array as shown here under:



local numbering global numbering

icon[0,0]=0
icon[0,1]=1
icon[0,2]=6
icon[0,3]=5

...

icon[6,0]=7
icon[6,1]=8
icon[6,2]=13
icon[6,3]=12

FEM in 2D
Exercise 3

A simple (time-dependent) analytical solution for the temperature equation exists for the case that the initial temperature field is

$$T(x,y,t=0) = T_0 + T_{max} \exp\left[-\frac{x^2+y^2}{\sigma^2}\right] \qquad (1)$$

where T_{max} is the maximum amplitude of the temperature perturbation at $(x,y)=(0,0)$ and σ its half-width.
The solution of the time-dependent PDE is

$$T(x,y,t) = T_0 + \frac{T_{max}}{1+4t\kappa/\sigma^2} \exp\left[-\frac{x^2+y^2}{\sigma^2+4t\kappa}\right] \qquad (2)$$

Set L_x=100km and $L_y = 80$km, $k=3$, $C_p = 1000$, $\rho = 3000$, $\tilde{Q}=0$, $T_{max} = 100^o$, $T_0 = 200^o$, and $\sigma = 10^4$m.
Generate a *nelx* $\times$ *nely* grid in the $[-L_x/2, L_x/2] \times [-L_y/2, L_y/2]$ domain. Write a function which takes x, y, t, T_0, T_{max}, κ and σ as argument and returns the analytical temperature value. Write a an explicit FEM code which solves the 2D diffusion equation. At each time step prescribe on the boundary the analytical solution.

We wish to compute the advection of a product-cosine hill in a prescribed velocity field. The initial temperature is:

$$T_0(x,y) = \begin{cases} \frac{1}{4}\left(1+\cos\pi\frac{x-x_c}{\sigma}\right)\left(1+\cos\pi\frac{y-y_c}{\sigma}\right) & \text{if } (x-x_c)^2+(y-y_c)^2 \leq \sigma^2 \\ 0 & \text{otherwise} \end{cases}$$

The boundary conditions are $T(x,y)=0$ on all four sides of the unit square domain. In what follows we set $x_c = y_c = 2/3$ and $\sigma = 0.2$.
The velocity field is analytically prescribed:
$\vec{v} = (-(y-L_y/2), +(x-L_x/2))$. Resolution is set to 31×31 nodes.
The timestep is set to $\delta t = 2\pi/200$ and we wish to carry out 200 timesteps so that the cone does a 2π rotation.
See Stone 43 for results/figures of this experiment obtained with Finite Elements.