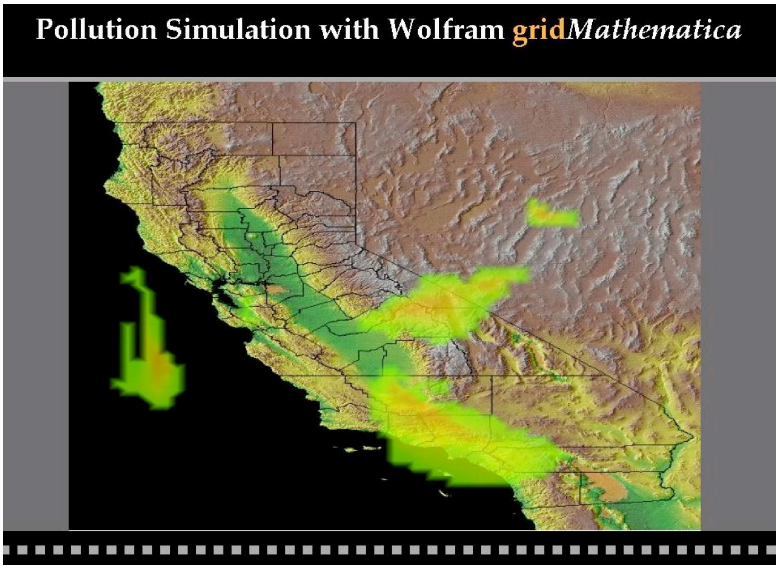
Large scale air pollution modeling with grid*Mathematica*

Anton Antonov
Wolfram Research, Inc.

Introduction

Out[2]=



- **Mathematical model**
- **Chemical schemes**
- **Advection simulation**
- **Two types of parallel algorithms**
- **Data feeding (using OOP)**
- **Development directions**

Mathematical model

An air pollution model is given by a system of PDE's:

$$\frac{\partial c_s}{\partial t} = -\frac{\partial (u\, c_s)}{\partial x} - \frac{\partial (v\, c_s)}{\partial y} - \frac{\partial (w\, c_s)}{\partial z} + \frac{\partial}{\partial x}\left(K_x \frac{\partial c_s}{\partial x}\right) +$$
$$\frac{\partial}{\partial y}\left(K_y \frac{\partial c_s}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_z \frac{\partial c_s}{\partial z}\right) + E_s + Q_s\left(c_1, c_2, \ldots, c_q\right) - (k_{1\,s} + k_{2\,s})\, c_s,$$

$s = 1, 2, \ldots, q.$

- the concentrations are denoted by c_s
- u,v,w and are wind velocities
- K_x,K_y, and K_z are diffusion coefficients
- the emission sources are described by E_s
- $k_{1\,s}$ and $k_{2\,s}$ are deposition coefficients
- the chemical reactions are described by the non-linear functions $Q\left(c_1, c_2, \ldots, c_q\right)$
- the number of equations q is equal to the number of species in the model

Europe covered with steer tanks

Numerical treatment of the mathematical model

Splitting the processes

It is hard to treat the mathematical model directly. Different kinds of splitting are used.

$$\frac{\partial c_s}{\partial t} = -\frac{\partial (u\, c_s)}{\partial x} - \frac{\partial (v\, c_s)}{\partial y},$$
$$\frac{\partial c_s}{\partial t} = \frac{\partial}{\partial x}\left(K_x \frac{\partial c_s}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial c_s}{\partial y}\right),$$
$$\frac{\partial c_s}{\partial t} = E_s + Q_s\left(c_1, c_2, \ldots, c_q\right),$$
$$\frac{\partial c_s}{\partial t} = -(k_{1\,s} + k_{2\,s})\, c_s,$$
$$\frac{\partial c_s}{\partial t} = -\frac{\partial (w\, c_s)}{\partial z} + \frac{\partial}{\partial z}\left(K_z \frac{\partial c_s}{\partial z}\right),$$

$s = 1, 2, \ldots, q.$

Reducing to 2D

In practice often the last system is skipped. The simulations discussed in this talk use this splitting:

$$\frac{\partial c_s}{\partial t} = -\frac{\partial (u\, c_s)}{\partial x} - \frac{\partial (v\, c_s)}{\partial y} + \frac{\partial}{\partial x}\left(K_x \frac{\partial c_s}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial c_s}{\partial y}\right),$$

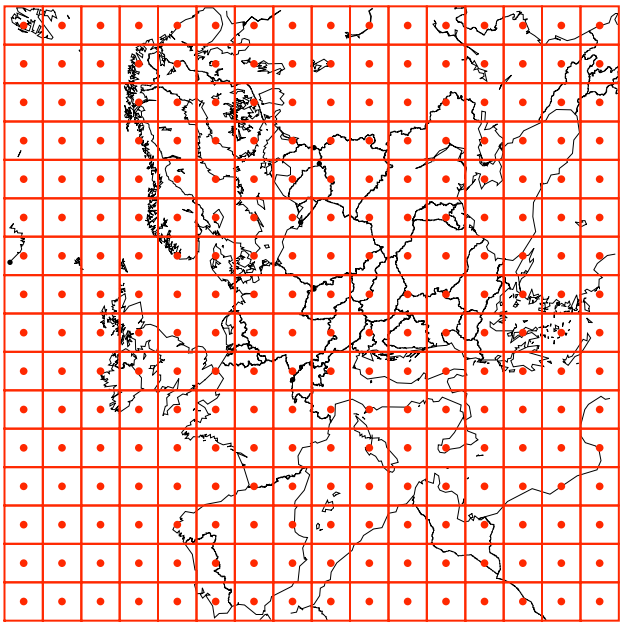
$$\frac{\partial c_s}{\partial t} = E_s + Q_s\left(c_1, c_2, \ldots, c_q\right),$$

$s = 1, 2, \ldots, q.$

Grid interpretation

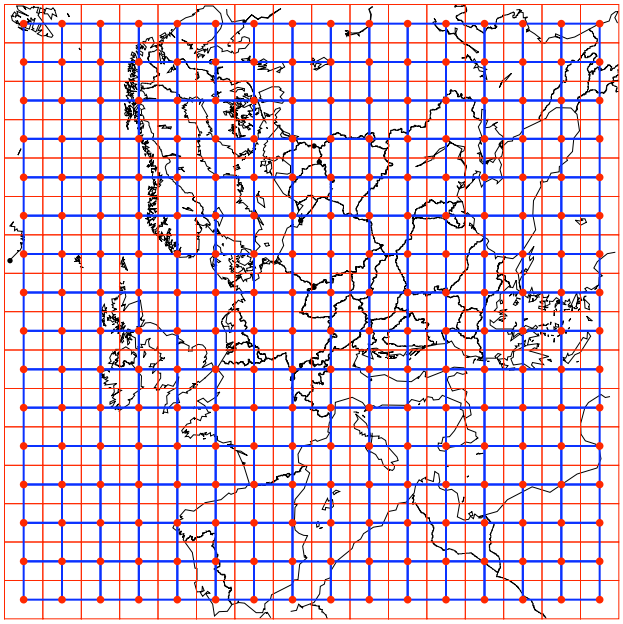
■ Steer tanks

Out[25]=



Simulation rectangular mesh

Out[27]=



code

Chemical schemes

Photo-chemical mechanisms

NO_x O_3 scheme

The following chemical air pollution mechanism is from
"*Athomospheric chemistry and physics*", J.H.Seinfeld & S.N.Pandis, 1998

Carbon monoxide athmospheric oxidation

$$\mathrm{CO} + \mathrm{OH} \xrightarrow{\mathrm{O_2}} \mathrm{CO_2} + \mathrm{HO_2}$$
$$\mathrm{HO_2} + \mathrm{NO} \longrightarrow \mathrm{NO_2} + \mathrm{OH}$$
$$\mathrm{NO_2} + \mathrm{h}\,\nu \longrightarrow \mathrm{NO} + \mathrm{O}$$
$$\mathrm{O} + \mathrm{O_2} + \mathrm{M} \longrightarrow \mathrm{O_3} + \mathrm{M}$$

■ Two cycles fast and slow

$$NO_2 \xrightarrow[H_2O]{h\nu} NO + O_3$$

$$CO + OH \xrightarrow[NO]{} HO_2$$

■ Equations for NDSolve

```
In[41]:= species = {"O3", "NO2", "NO", "CO"};

In[42]:= a = 1 / (1 + k5.22 c["M"] / (k5.23 c["H2O", RH]))

Out[42]= 0.0364314

In[43]:= eq["O3"] = c["O3", t] == k5.1[t] c["NO2", t] / (k5.3 c["NO", t] + k5.21b a)

In[44]:= eq["NO2"] = D[c["NO2", t], t] == k5.1[t] k5.21b a (2 k5.24 c["CO", t] / k5.26 - 3 c["NO2", t]) / (k5.3 c["NO", t] + k5.21b a)

In[45]:= eq["NO"] = D[c["NO", t], t] == j5.1 k5.21b a (c["NO2", t] - 2 k5.24 c["CO", t] / k5.26) / (k5.3 c["NO", t] + k5.21b a)

In[46]:= eq["CO"] = D[c["CO", t], t] == 2 k5.1[t] k5.21b k5.24 a c["CO", t] / k5.26 / (k5.3 c["NO", t] + k5.21b a)

In[47]:= c["M"] = 780 840 + 209 460; (* N2+O2 ppm *)

In[48]:= c["M"]

Out[48]= 990 300
```

■ Using QSSA

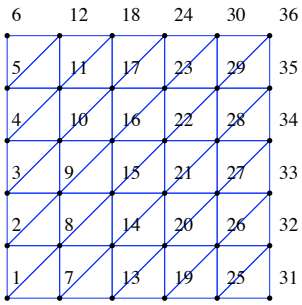
With QSSA we have algebraic equations only. (No solving. Technically speaking, an $O(0)$ approximation.)

Advection simulation

The following schemes were implemented:

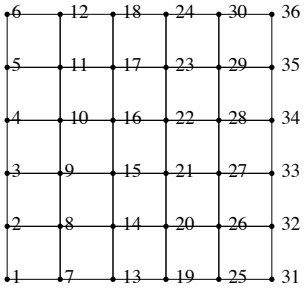
- **1D pseudo spectral**
- **1D finite elements**
  - **Locally 1D finite elements implementation**
- **2D finite elements**
  - **Using triangle finite elements**

Out[11]=



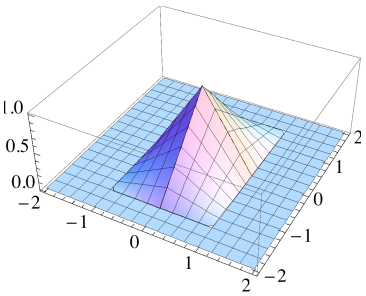
  - **Using square finite elements**
    - ***grid***

Out[41]=



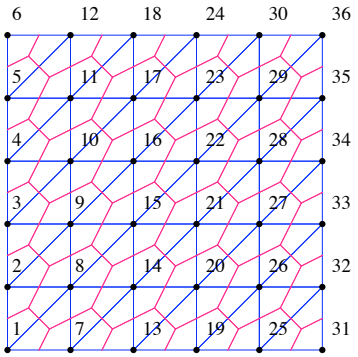
    - ***basis function***

Out[52]=



2D control volume elements

Out[116]=



2D×1D finite elements

Time step with Crank-Nicholson

From the finite element discretization we have the following matrix equation:

$$P \frac{\partial g}{\partial t} = -A\, g.$$

The following is the Crank-Nicholson approximation:

$$g(t_n + \triangle t) = -P^{-1} A\, g(t_n) + \int_{t_n}^{t_{n+1}} P^{-1} A\, g(s)\, ds,$$

$$g(t_n + \triangle t) \approx -P^{-1} A\, g(t_n) + \frac{P^{-1} A\, g(t_{n+1}) - P^{-1} A\, g(t_n)}{2\, \triangle t}$$

From the general Crank-Nicholson form:

$$g_{n+1} = g_n - \triangle t \left(\theta\, P^{-1} A\, g_{n+1} + (1-\theta)\, P^{-1} A\, g_n\right),$$

we derive the computational form:

$$\left(I + \triangle t\, \theta\, P^{-1} A\right) g_{n+1} = \left(I - \triangle t (1-\theta)\, P^{-1} A\right) g_n.$$

(How many chemical steps per advection step?)

Interpretation of the equations

Since our implementation is with *Mathematica* we can easily see the semi-discrete equations of the numerical scheme.

In[263]:=
```
G = Table[g[t, i], {i, 1, Length[Nodes /. meshObject]}];
```

In[264]:=
```
Thread[KMatrix.D[G, t] == AMatrix.G][[12]]
```

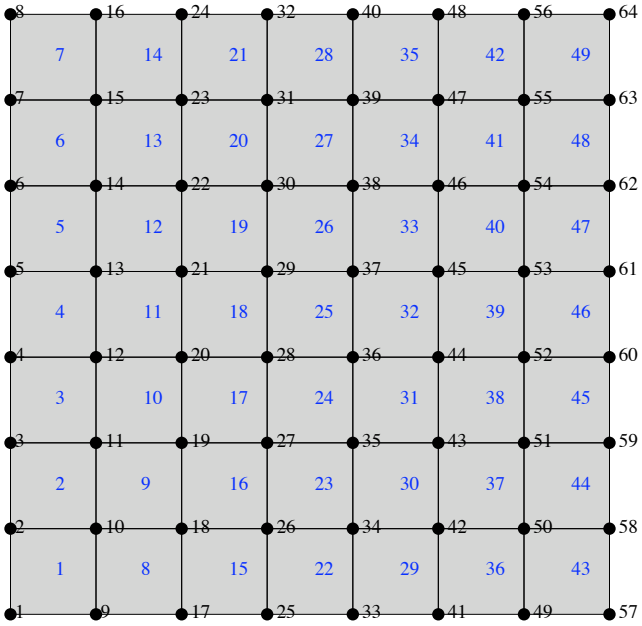
Out[264]=
$$0.000434028\, g^{(1,0)}[t, 3] + 0.00173611\, g^{(1,0)}[t, 4] + 0.000434028\, g^{(1,0)}[t, 5] + 0.00173611\, g^{(1,0)}[t, 11] + 0.00694444\, g^{(1,0)}[t, 12] + 0.00173611\, g^{(1,0)}[t, 13] + 0.000434028\, g^{(1,0)}[t, 19] + 0.00173611\, g^{(1,0)}[t, 20] + 0.000434028\, g^{(1,0)}[t, 21] == (0.333333\, d + 0.0104167\, u + 0.0104167\, v)\, g[t, 3] + \left(0.333333\, d + 0.0416667\, u - 1.73472 \times 10^{-18}\, v\right) g[t, 4] + (0.333333\, d + 0.0104167\, u - 0.0104167\, v)\, g[t, 5] + \left(0.333333\, d - 1.73472 \times 10^{-18}\, u + 0.0416667\, v\right) g[t, 11] + \left(-2.66667\, d - 1.04083 \times 10^{-17}\, u - 1.04083 \times 10^{-17}\, v\right) g[t, 12] + (0.333333\, d + 0.\, u - 0.0416667\, v)\, g[t, 13] + (0.333333\, d - 0.0104167\, u + 0.0104167\, v)\, g[t, 19] + (0.333333\, d - 0.0416667\, u + 0.\, v)\, g[t, 20] + (0.333333\, d - 0.0104167\, u - 0.0104167\, v)\, g[t, 21]$$

In[266]:=
```
Thread[KMatrix.D[G, t] == AMatrix.G][[64]] // Chop

CoefficientArrays[%[[2]], {u, v, d}] // Normal
```

```
{0, {0.0104167 g[t, 55] + 0.0208333 g[t, 56] - 0.0104167 g[t, 63] - 0.0208333 g[t, 64],
  0.0104167 g[t, 55] - 0.0104167 g[t, 56] + 0.0208333 g[t, 63] - 0.0208333 g[t, 64],
  0.333333 g[t, 55] + 0.166667 g[t, 56] + 0.166667 g[t, 63] - 0.666667 g[t, 64]}}
```

Out[265]=



■ Anisotropy of the schemes

Consider the equation

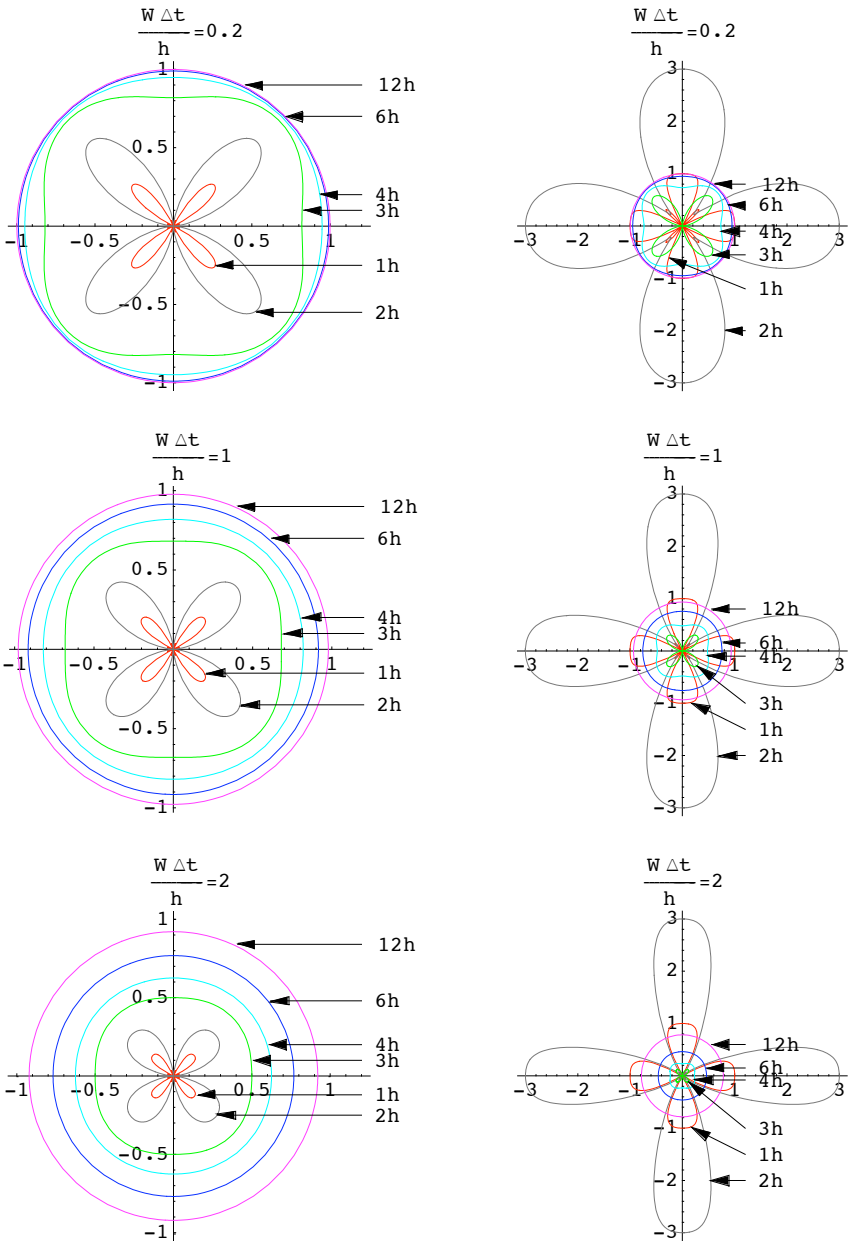
$$\frac{\partial c}{\partial t} = -W \cos\phi \frac{\partial c}{\partial x} - W \sin\phi \frac{\partial c}{\partial y}$$

The **phase velocity** of a wave is the rate at which the phase of the wave propagates in space.

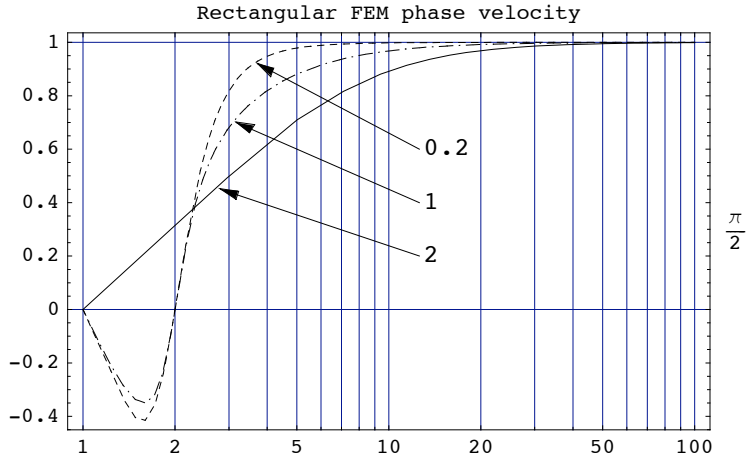
The **group velocity** of a wave is the velocity with which the variations in the shape of the wave' s amplitude (modulation, envelope) propagate through space.

As Lighthill explains "... Perhaps the most fundamental property of the group velocity can be explained at once ... The energy of sinusoidal waves is propagated not at the wave speed W, but at the group velocity V".

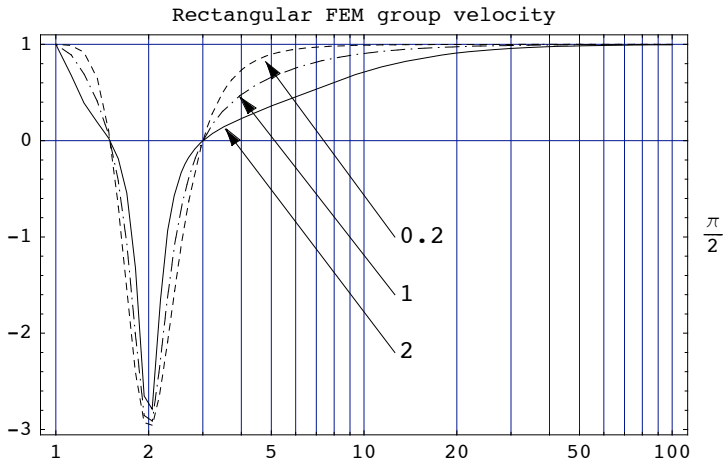
Phase and group velocities for a square FEM grid for various CFL numbers, $\frac{W\,\Delta t}{h}$



Phase velocity log linear plot for $\phi = \pi/2$



Group velocity log linear plot for $\phi = \pi/2$

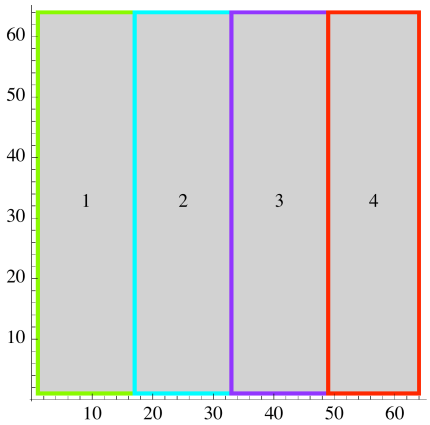


Two types of parallel algorithms

■ Parallelizing the physical domain

The domain can be partitioned among 4 processors across the x-axis in the following way:

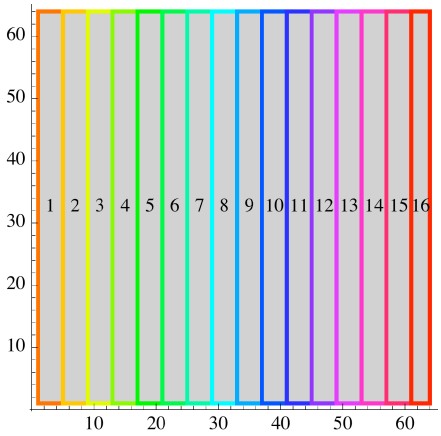
Out[39]=



The domains overlap. So basically, if FEM is used, we will run 4 FEM advection simulations in parallel. We may say that the parallelism is at the FEM level.

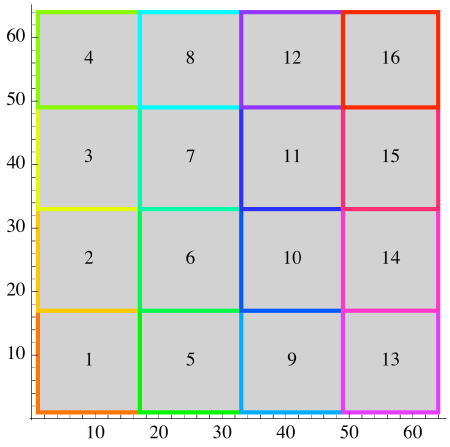
What will be the partition among 16 processors across the x-axis?

Out[36]=



Partitioning on both axes leads to a better conditioned space domains.

Out[37]=



Parallel advection test

What partitioning of the physical domain should be used?

-- if the grid has refinements?

-- or the processors have different computational abilities?

■ Parallelizing in the pseudo-space domain

It is easier to make parallel distributions of vectors (in Hilbert spaces).

If FEM is used this would mean that only one FEM is used for the advection simulation. The parallelism is pushed at the linear algebra level.

Data feeding (and OOP)

■ Dynamic and Static polymorphism

Design Patterns in Mathematica

■ The WindField object

Wind field data objects, through static inheritance and decoration.

- **WindField**
- **WindFieldEnhanced**
- **WindFieldSubRegion**

■ WindField object Test

In[322]:=
```
wf = WindFieldEnhanced["/home/antonov/OODEM/data/metdata/data1995/jul95winSpaced"];
wf@"ReadFile"[]
wf@"PartitionSize"
wf@"SetTimeStep"[21 600]
wf@"SetSpaceStep"[200 * 10^3]
```

Out[324]= 32

Out[325]= 21 600

Out[326]= 200 000

In[327]:=
```
wfsr = WindFieldSubRegion[wf, {{31, 64}, {1, 33}}]@"Create"[]
```

Out[327]=
```
WindFieldSubRegion[
 WindFieldEnhanced[/home/antonov/OODEM/data/metdata/data1995/jul95winSpaced]]
```

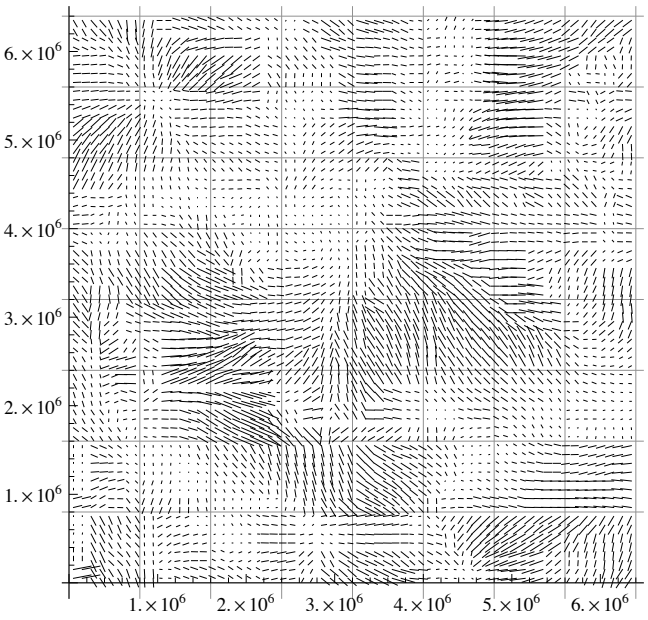
In[328]:=
```
{{x0, x1}, {y0, y1}} = wfsr@"SubregionCoordinates"
```

Out[328]= {{31, 64}, {1, 33}}

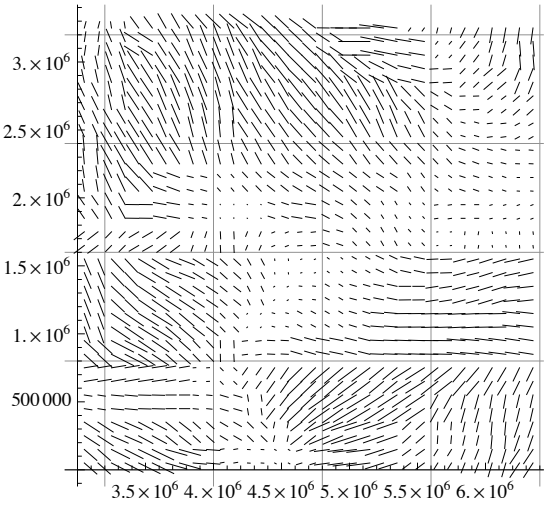
In[311]:=
```
{windx64, windy64} = wf@"InterpolateField"[100, 64, 64];
```

In[312]:=
```
{windx, windy} = wfsr@"InterpolateField"[100, 64, 64];
```

Out[316]=



Out[321]=



■ The AdvectionOperator object

Instead of assembling the advection matrix at each time step, we can derive as a linear combination of matrices assembled at the data time points.

$$A(t_n) = \theta\, A(t_{(k+1)\,m}) + (1-\theta)\, A(t_{k\,m}),$$

$$k\,m \le n \le (k+1)\,m,$$

$$\theta = \frac{t_n - t_{k\,m}}{t_{(k+1)\,m} - t_{k\,m}}.$$

■ AdvectionOperator

■ AdvectionOperator object Test

Parallel runs

■ Implementation studies

Parallel run

■ Profiles

IDF demo profiles

PSC1 - 4 slaves, for a total of 32 cores.
Each machine was linked via Mellanox Infiniband.

8 cores : 1 - step : 29 seconds
10 - steps : 515 seconds

16 cores : 1 - step : 17 seconds
10 - steps : 184 seconds

32 cores : 1 - step : 10 seconds
10 - steps : 105 seconds

Development directions

Larger chemical schemes

3D advection -- done for SC'06

Grid refinement -- done for both 2D and 3D

Summary

For this project were used the following *Mathematica* functions and features:

Piecewise with NIntegrate/Integrate,
NDSolve and its framework,
SparseArray's,
LinearSolve and its preconditioners,
OOP with Downvalues/Subvalues,
grid*Mathematica.*

Acknowledgements: Schoeller Porter, Manjula Jayasuriya.