Case Study 1: Atmospheric Model

Conservation of momentum:
\[
\frac{du}{dt} - \left( f + u \frac{\tan \phi}{a} \right) v = -\frac{1}{a \cos \phi} \frac{1}{\rho} \frac{\partial p}{\partial \lambda} + F_\lambda
\]
\[
\frac{dv}{dt} + \left( f + u \frac{\tan \phi}{a} \right) u = -\frac{1}{\rho a} \frac{\partial p}{\partial \phi} + F_\phi
\]

Hydrostatic approximation:
\[
g = -\frac{1}{\rho} \frac{\partial p}{\partial z}
\]

Conservation of mass:
\[
\frac{\partial p}{\partial t} = -\frac{1}{a \cos \phi} \left( \frac{\partial}{\partial \lambda} (\rho u) + \frac{\partial}{\partial \phi} (\rho v \cos \phi) \right) - \frac{\partial}{\partial z} (\rho w)
\]

Conservation of energy:
\[
C_p \frac{dT}{dt} - \frac{1}{\rho} \frac{dp}{dt} = Q
\]

State equation (atmosphere):
\[
p = \rho RT
\]
Case Study 1: Atmospheric Model (cont)

“Atmosphere in a box”:

Derivatives grid stencil:

- 9 points in horizontal
- 3 points in vertical

Other necessary operations:

\[
\text{Total Mass} = \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} \sum_{k=0}^{N_z-1} M_{ijk}.
\]

Other processes (“extra physics” = precipitation, clouds, radiation):

SAY, LOTS OF THEM AND LOTS OF COMPUTATION
Complicated accumulations of all z levels only, no x dependence

\[
TCS_k = \prod_{i=1}^{k} (1 - \epsilon \delta_i) TCS_1
\]

\[
= TCS_{k-1} (1 - \epsilon \delta_k).
\]

Student’s task:

Go through algorithmic design process (PCAM) for this problem …
Arranging electronic components on a chip.

Real estate is a big issue => Minimise total area.

Components have to be in a certain order (to connect, heat etc; e.g. above and to left of ...)

But might be able to have various configurations.

Treat as “cells” of specific volume, various allowable configurations, order specified.

⇒ Search through set of all possible solutions to minimise the total area.

First convince yourself that this is a search algorithm by finding the answer by hand.

HOMEWORK:

First convince yourself that this is a tree search algorithm by finding the answer by hand.

Bear in mind that the exhaustive search can be considerably improved by “pruning” i.e. in this “depth-first” algorithm, stop if branch has no chance of providing min area (area already greater than best solution so far).

Convince yourself that the pseudo-code for “branch-and-bound” algorithm on the next page is what you are actually doing.

Consider how to design a parallel algorithm for this problem (PCAM).
Logic:

```
procedure b&b_search(A )
begin
    A_min = \infty
    b&b_search_1(A )
end

procedure b&b_search_1(A )
begin
    score = eval(A )
    if(score < A_min) then
        if (leaf(A )) then
            A_min = score
            report solution and score
        else
            foreach child A_i of A
                b&b_search_1(A_i)
            endfor
        endif
    endif
end
```

Students: Discuss PCAM for this problem ...