

# Speeding up single-particle tracking in Geant and GeantV using vectorisation

John Apostolakis, CERN



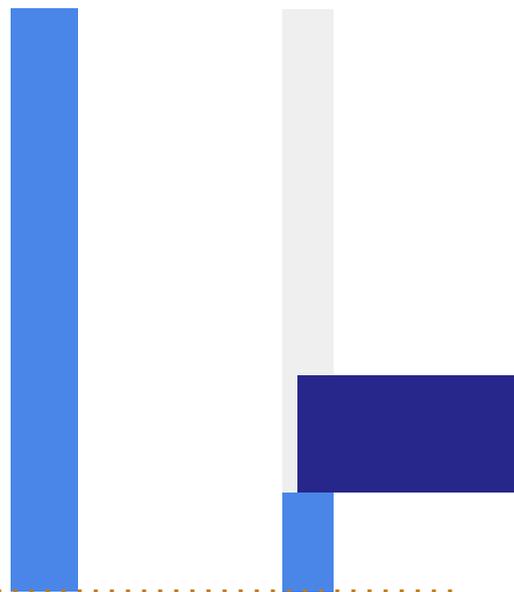
# The challenge

In Geant4 particles are always tracked one at a time.

In GeantV non-trivial fraction,  $O(10\%)$  is currently tracked in single particle mode

- volumes with small flux or volume
- 'straggler' tracks to end event(s)

Fraction of work in single particle mode



Geant4

GeantV

A laptop screen is shown in a dark, dimly lit environment. The screen displays a data dashboard. At the top, there is a line graph with a blue line and several data points, showing an overall upward trend. Below the graph, there is a pie chart with a blue and green segment. The text 'New York' is visible next to the pie chart. The laptop keyboard is partially visible at the bottom of the frame.

# Components with vectorisation potential



**The  
'candidates'**

**VecGeom** is an existing common component. Two parts suit vectors:

- Solids with several similar faces
- Intersection of bounding boxes
- Intersection of same solids\*

Runge Kutta **integration** of charged particle trajectories

- integrates 6-8 variables at once

Others?



# Geometry - Solids

Solids with vector code for one 'track': LIST HERE

## Step 1

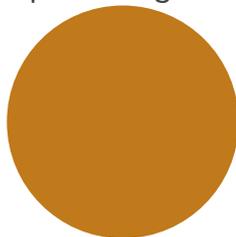
Identify solids with most vectorisation potential



? Trap, Polycone, Polygon?

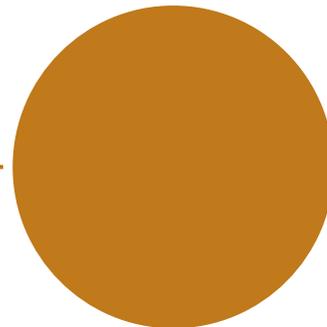
## Step 2

Implementation of key methods (intersection, safety) of 1-2 most promising solid(s)



## Step 3

Benchmark and extend to additional candidates types of solids





## Vectors in 'scalar' Navigation

VecGeom Navigator vectorizes over bounding boxes of candidate volumes

- first results reported in a previous Geant4 meeting
- Good speedups (2-3) in volumes with many daughters, no speedup in case of 0-1 daughters!

Trial integration for navigation in G4 is under study.



# Runge Kutta integration

Intertwines

- Vector operations over 6 (x,p) or 8 (x, p, E, t) variables
- calls to field evaluation and force calculation

Using templated 'calls' to inline field methods, all the code can be put into one compilation unit

- Prototype in G4 by Josh Xie (GSoC '14), reused in GeantV

Plan to investigate vectorisation potential for scalar tracks - potential GSoC project in 2018

$$f_1 = f(y_0 + c_1 f_0 s)$$

$$b_2 = a_{20} f_0 + a_{21} f_1$$

$$f_2 = f(y_0 + c_2 f_0 s)$$

$$b_3 = a_{30} f_0 + a_{31} f_1 + a_{32} f_2$$

$$f_3 = f(y_0 + c_3 f_0 s)$$

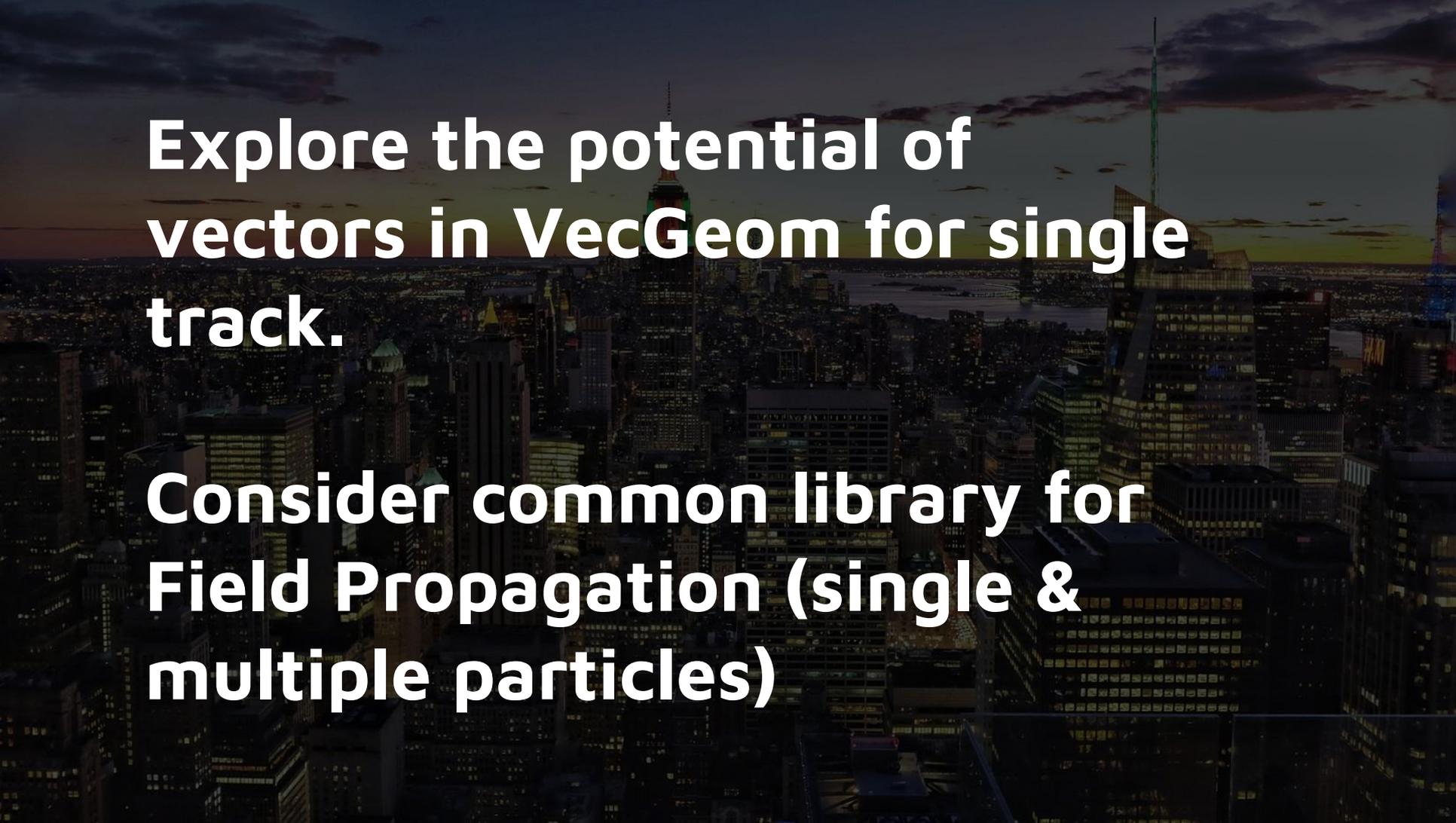
$$b_4 = a_{40} f_0 + a_{41} f_1 + a_{42} f_2 + a_{43} f_3$$

$$\dots$$

$$y_{\text{end}} = y_0 + s * \square d_i f_i$$

$$y_{\text{error}} = \square (d_i - d_i^*) f_i$$

where 'b<sub>i</sub>' & 'v<sub>i</sub>' vectors of 6-8 values  
 a<sub>ij</sub> are constants of the method  
 f is 'force' = q ( E + q x v )

An aerial view of a city skyline at dusk, with a semi-transparent dark overlay. The text is centered and reads: 

**Explore the potential of  
vectors in VecGeom for single  
track.**

**Consider common library for  
Field Propagation (single &  
multiple particles)**