

IDL

Vectorisation

FOR Loops V Array Ops

- FOR loops are slow because of overheads (e.g. repeated indexing)

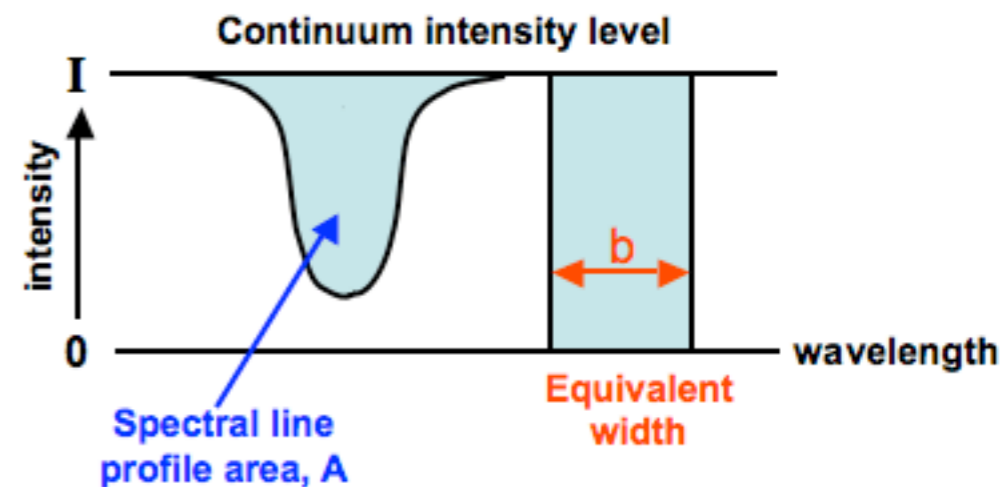
Example: Sum of uniform deviates

http://www.ast.cam.ac.uk/~vasily/idl/total_loop.pro

- Hence, reduce the number of iterations and do more at each iteration
- This can be done with *vectorisation*
- Caution: (a lot of) memory is needed

Case Study: Equivalent Width Calculation

- For a sample of N_{data} spectra each containing N_{pix} flux measurements, calculate equivalent width of H α absorption line.



- FOR loops version

http://www.ast.cam.ac.uk/~vasily/idl/eqwidthI_forloops.pro

- Vectorised version

http://www.ast.cam.ac.uk/~vasily/idl/eqwidthI_vectorised.pro

What if there is IF?

- Use WHERE to create *index* of the array
- `index = where(array expression)`
- `array[where]` or `(array_expression)[index]`
- Note: array index is ALWAYS one-dimensional regardless of the dimensionality of the original array

http://www.ast.cam.ac.uk/~vasily/idl/eqwidth2_vectorised.pro

Where you don't need WHERE

- $a = \text{sqrt}(b[\text{where}(b > 0)])$

compare to

- $a = \text{sqrt}(b > 0)$