Flexi-Futures

Jeremy Singer

Outline

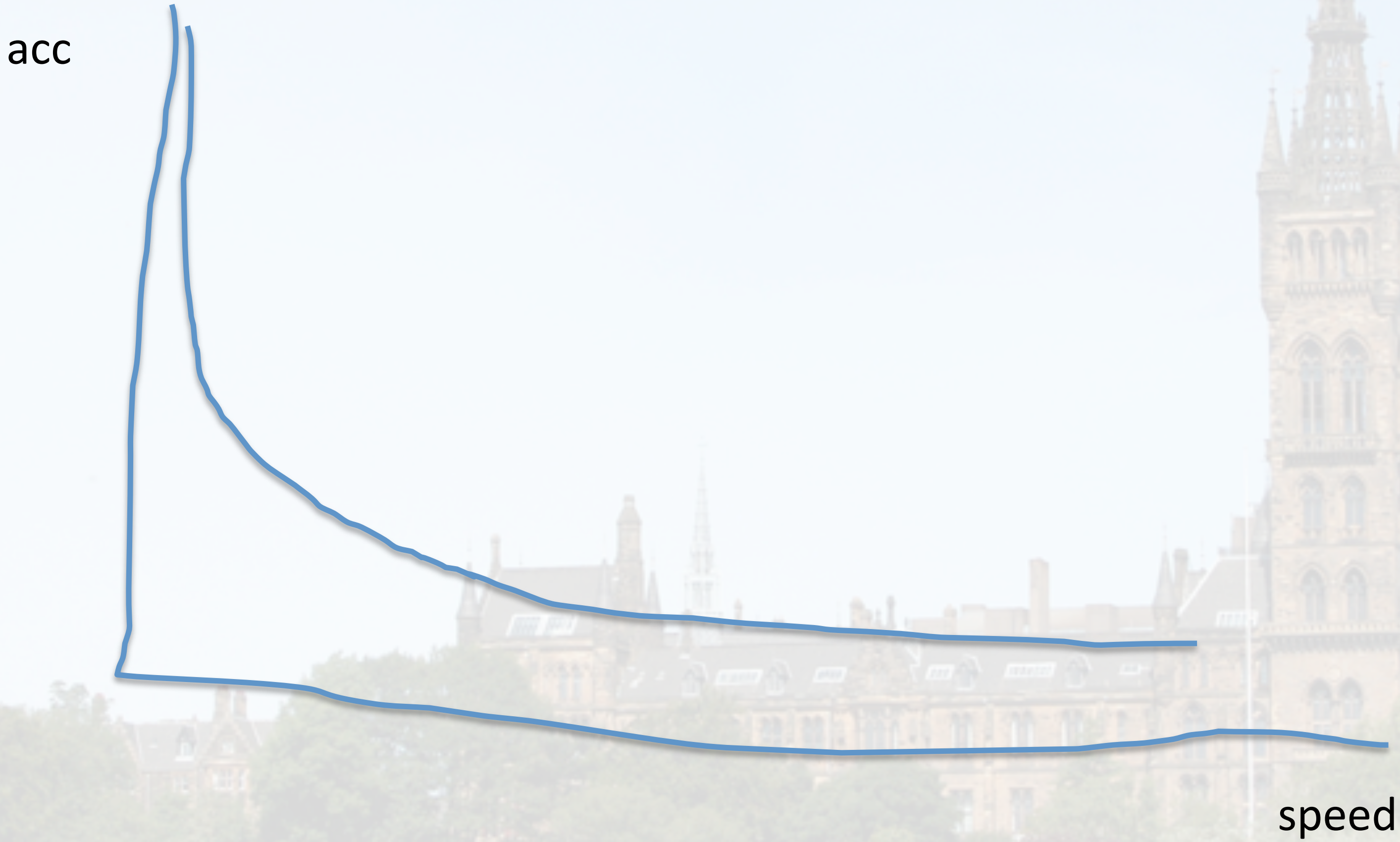
1. Runtime tradeoffs
2. Support for flexi-futures
3. Applications

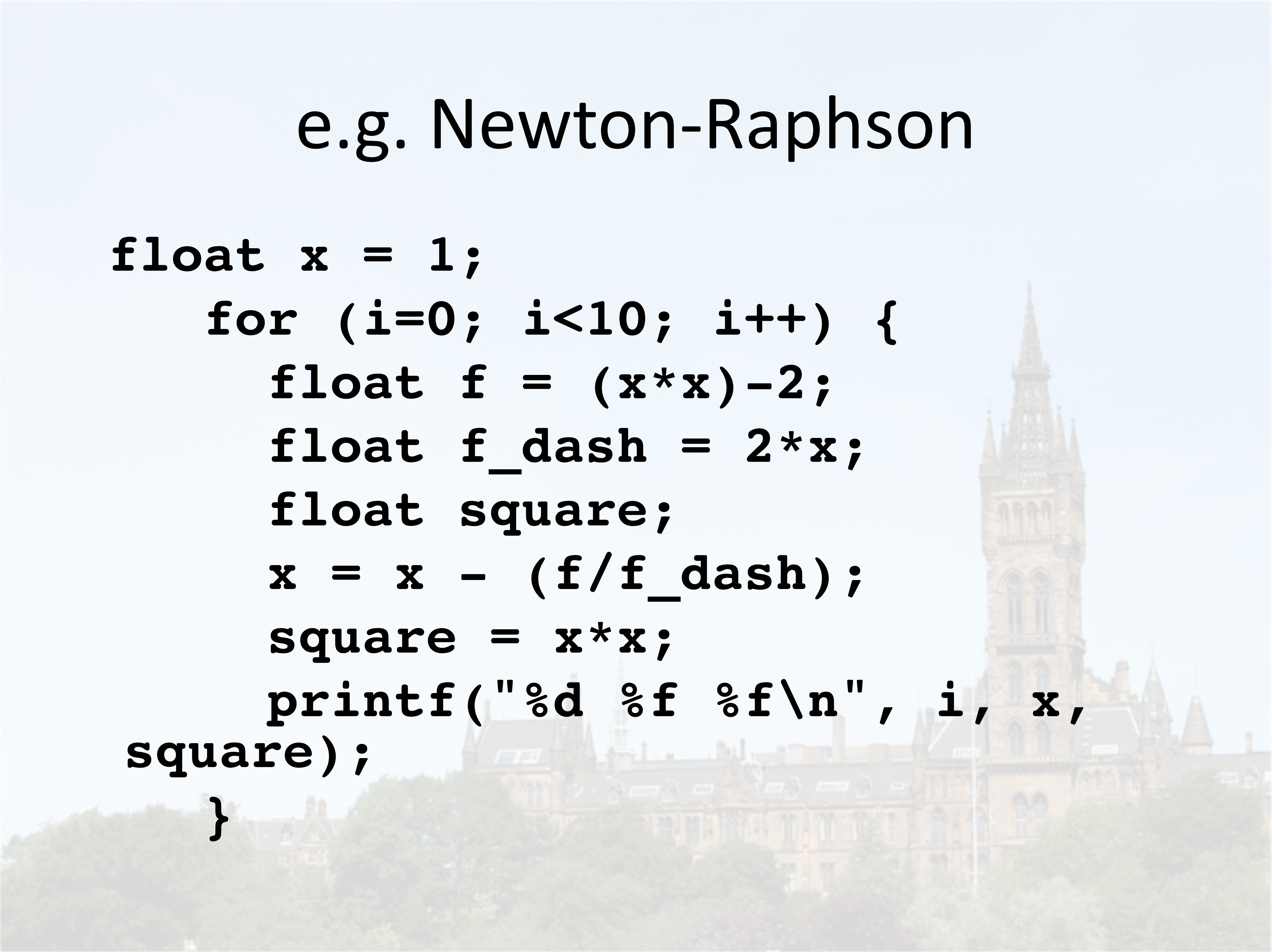
Outline

1. Runtime tradeoffs
2. Support for flexi-futures
3. Applications

Tradeoff #1

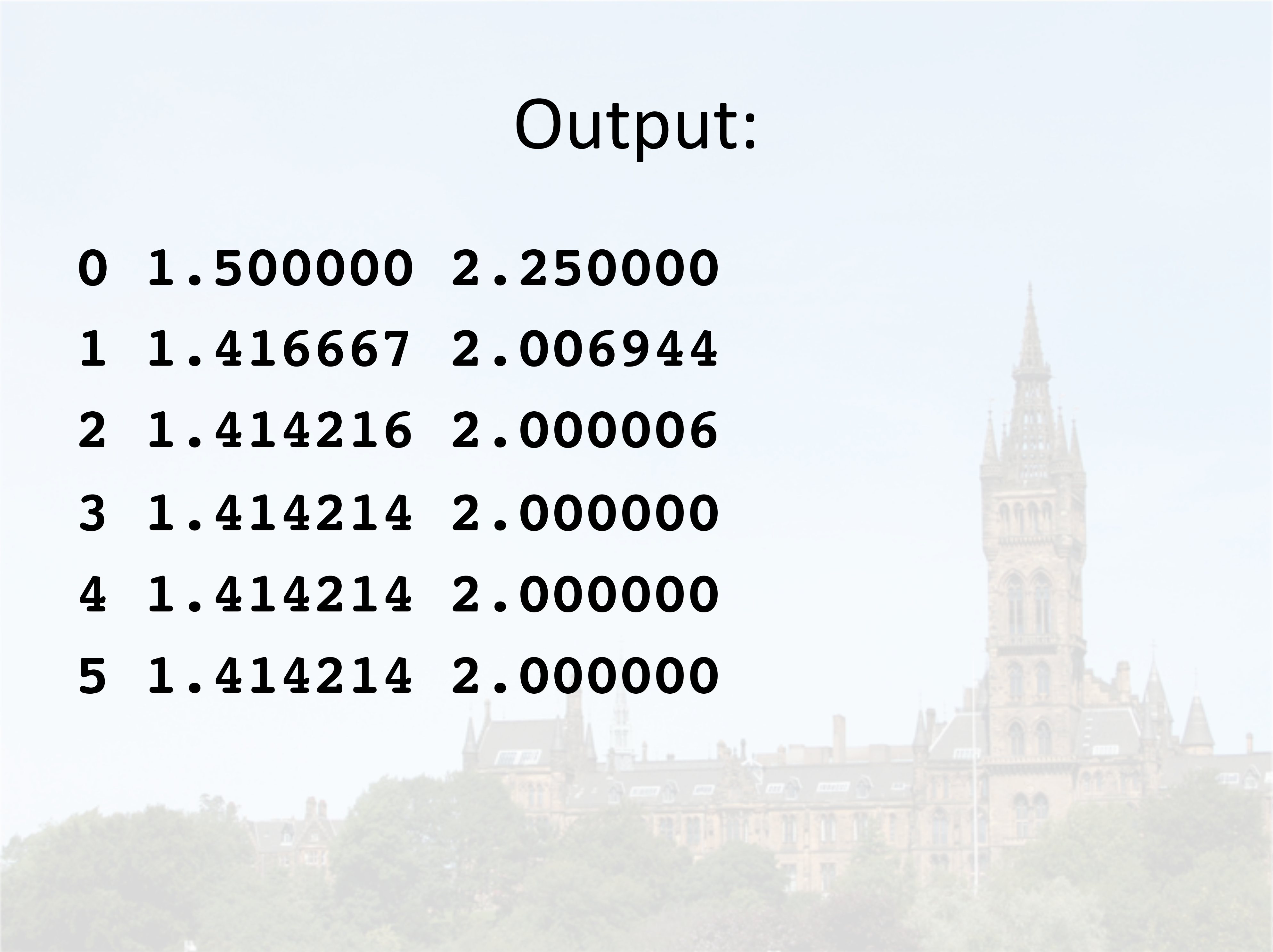
- accuracy versus speed





e.g. Newton-Raphson

```
float x = 1;
   for (i=0; i<10; i++) {
     float f = (x*x)-2;
     float f_dash = 2*x;
     float square;
     x = x - (f/f_dash);
     square = x*x;
     printf("%d %f %f\n", i, x,
square);
   }
```

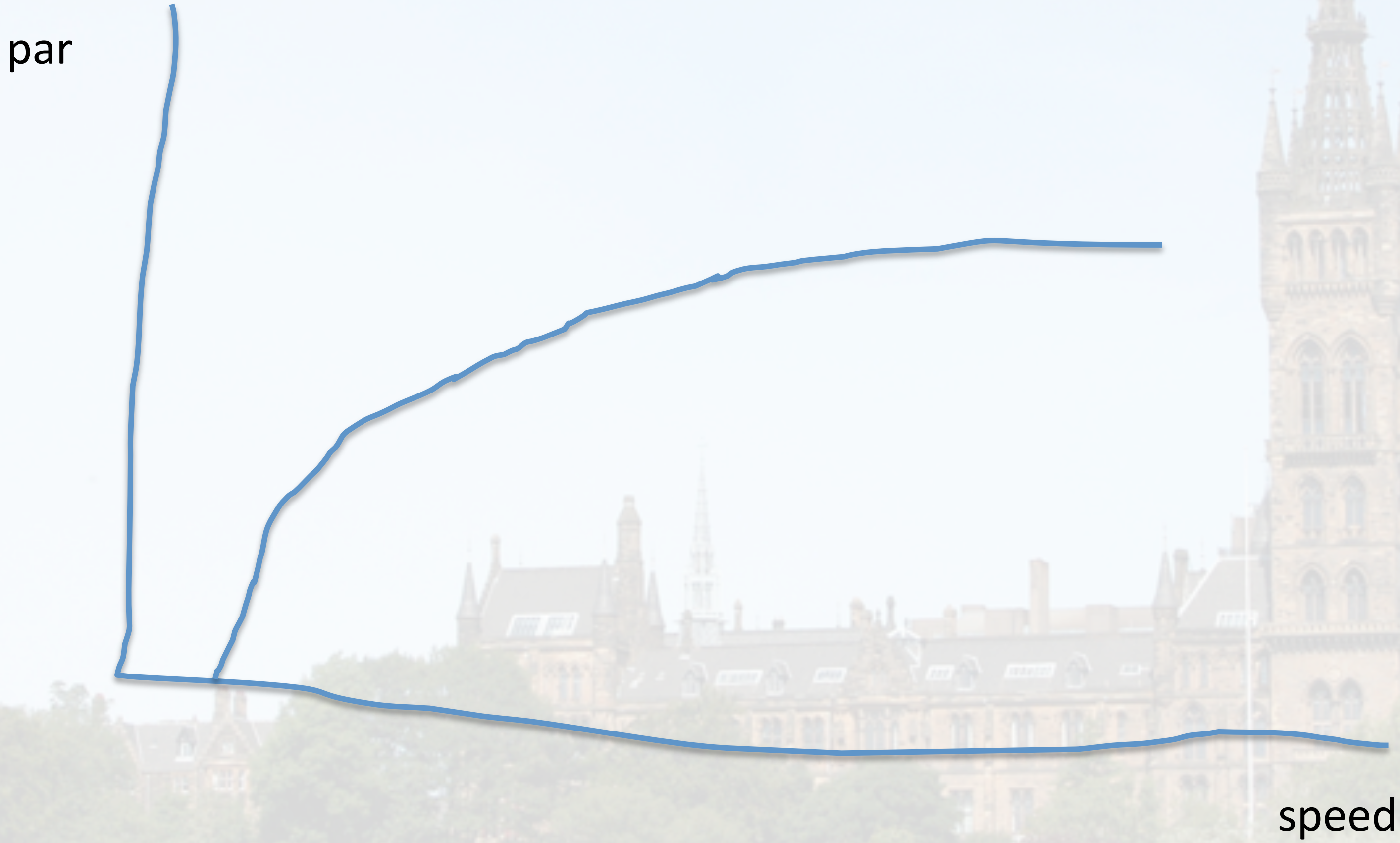


Output:

```
0 1.500000 2.250000
1 1.416667 2.006944
2 1.414216 2.000006
3 1.414214 2.000000
4 1.414214 2.000000
5 1.414214 2.000000
```

Tradeoff #2

- parallelism versus speed

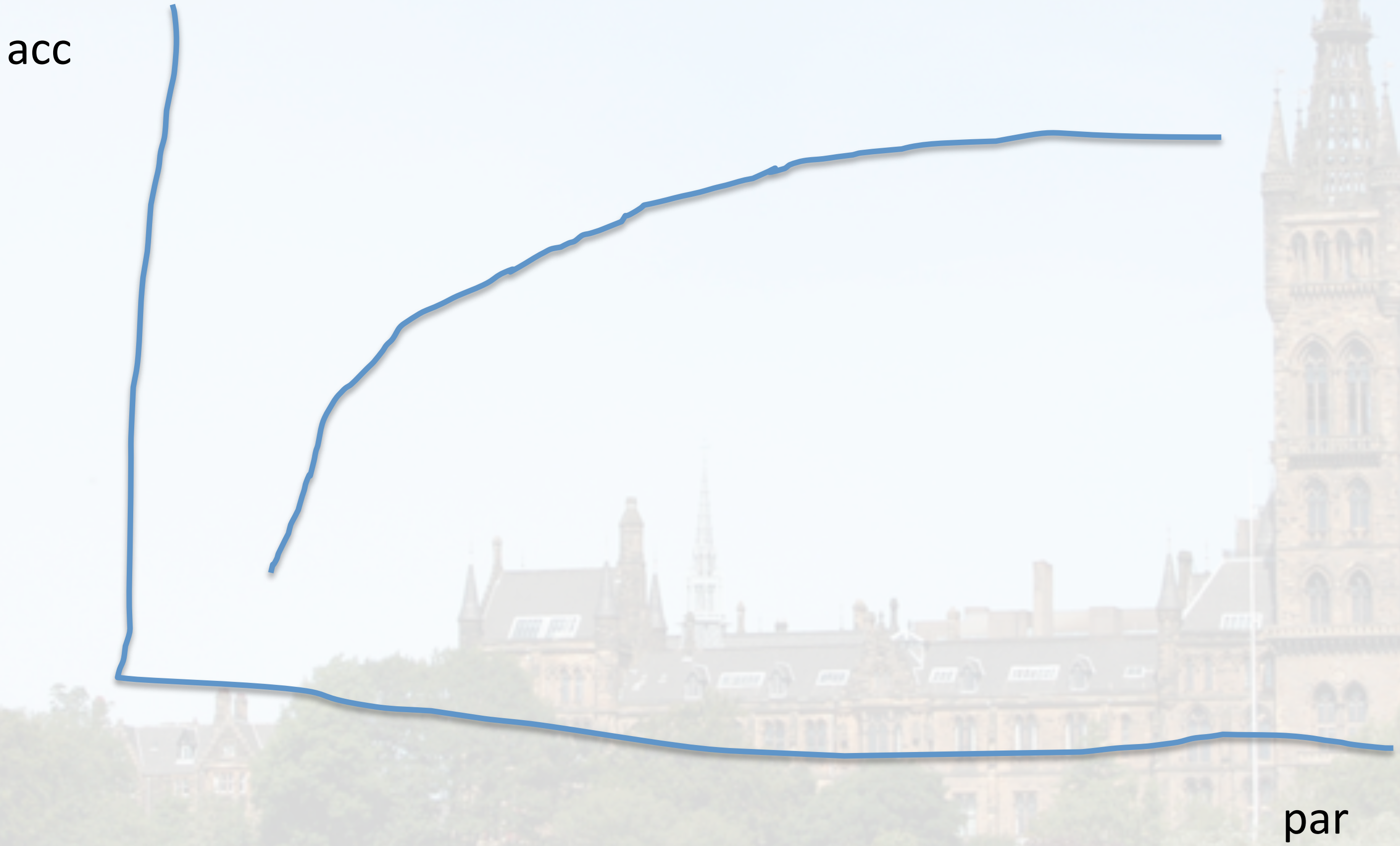


nBody simulation

```
for(int i=0; i < bodies.length; ++i)
  {           Body iBody = bodies[i];         for
  (int j=i+1; j < bodies.length; ++j)
  {               double dx = iBody.x - bodies
  [j].x;           double dy = iBody.y - bodies
  [j].y;           double dz = iBody.z - bodies
  [j].z;               double dSquared = dx * dx +
  dy * dy + dz * dz;                double distance
  = Math.sqrt(dSquared);                double mag =
  dt / (dSquared * distance);            iBody.vx -=
  dx * bodies[j].mass * mag;             iBody.vy -=
  dy * bodies[j].mass * mag;             iBody.vz -=
  dz * bodies[j].mass * mag;             bodies[j].vx
  += dx * iBody.mass * mag;             bodies[j].vy
  += dy * iBody.mass * mag;             bodies[j].vz
  += dz * iBody.mass * mag;         }      }
```

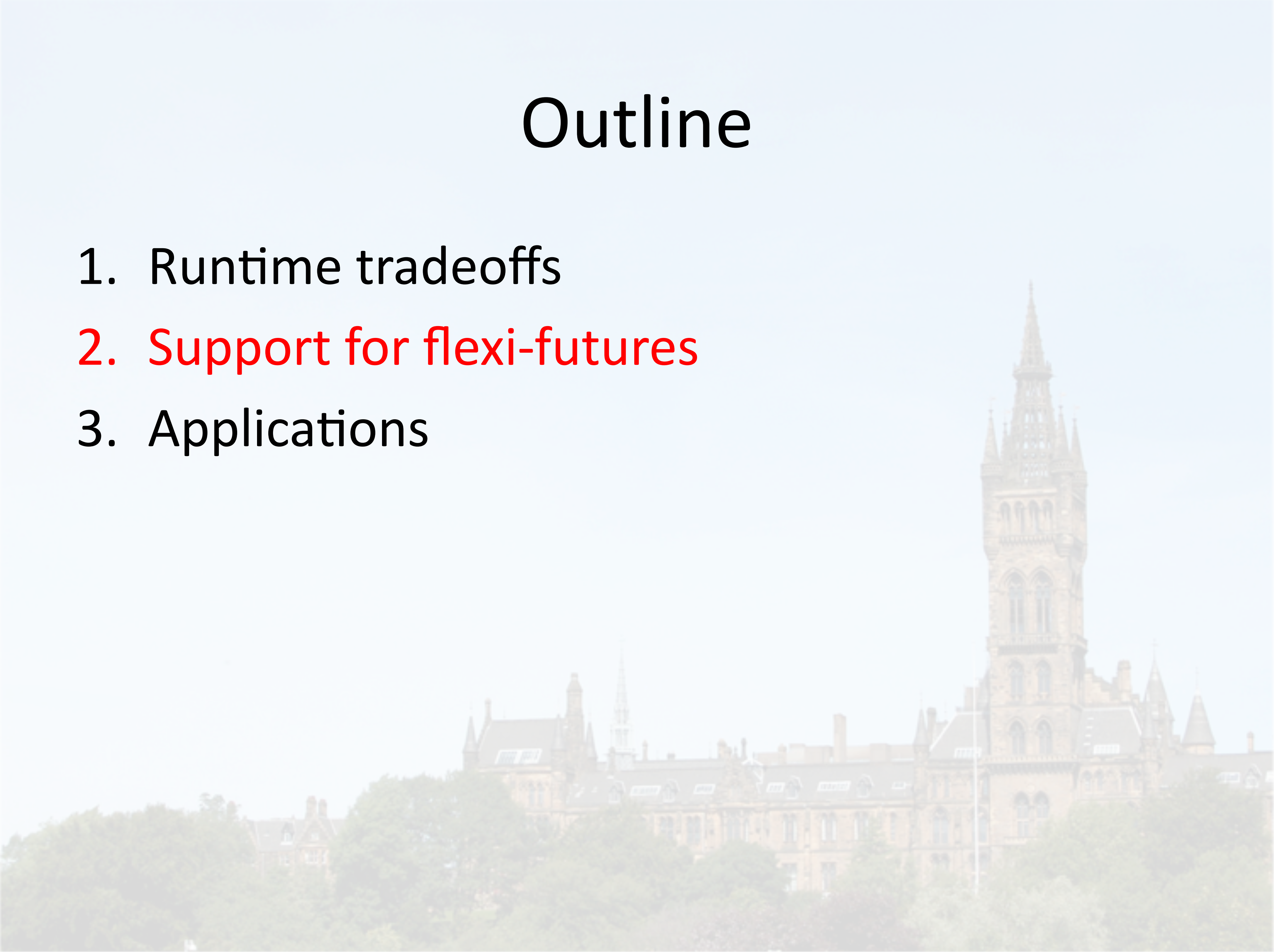
Tradeoff #3

- parallelism versus accuracy



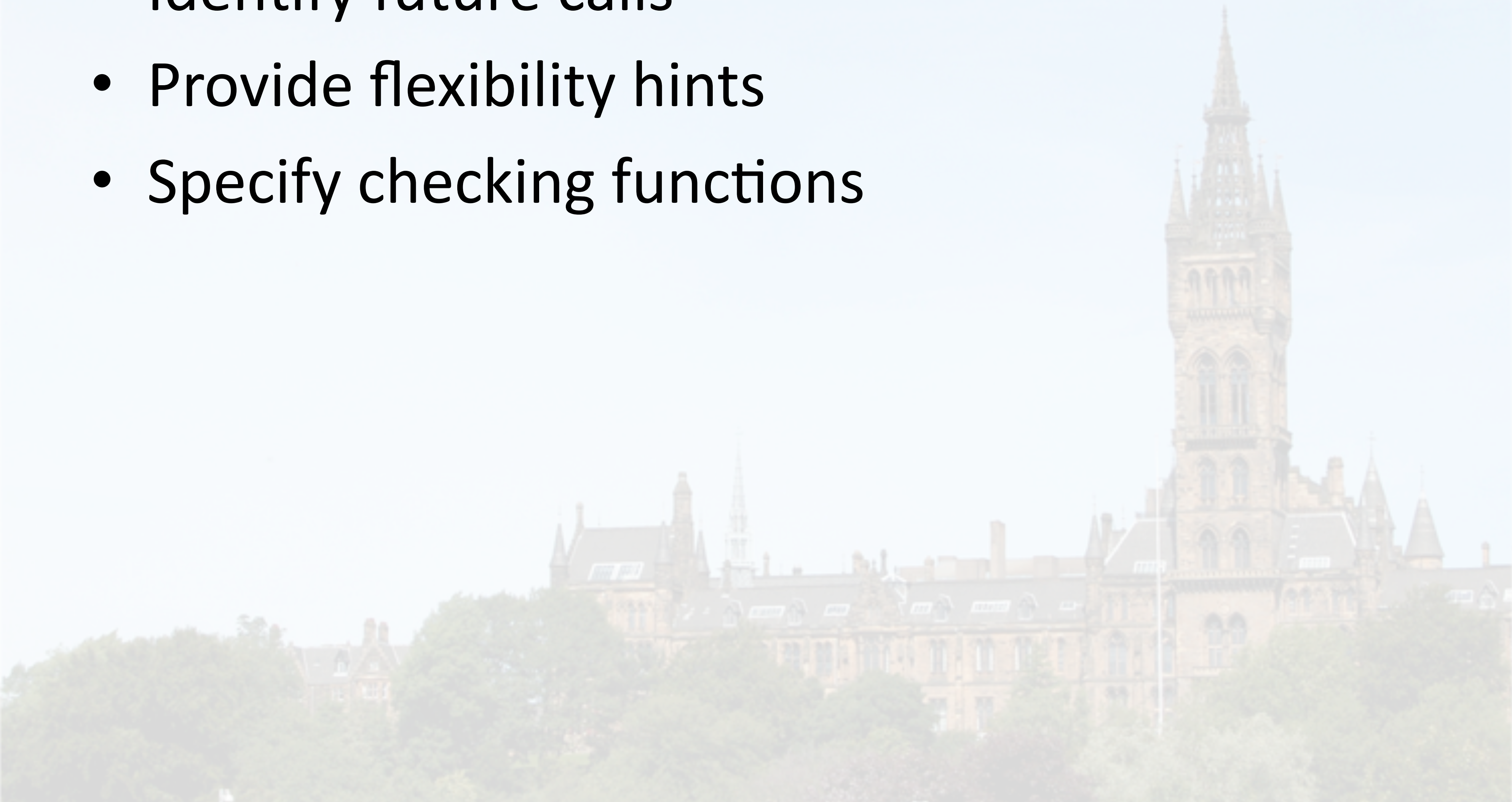
Outline

1. Runtime tradeoffs
2. Support for flexi-futures
3. Applications



ProgLangs support

- Identify future calls
- Provide flexibility hints
- Specify checking functions

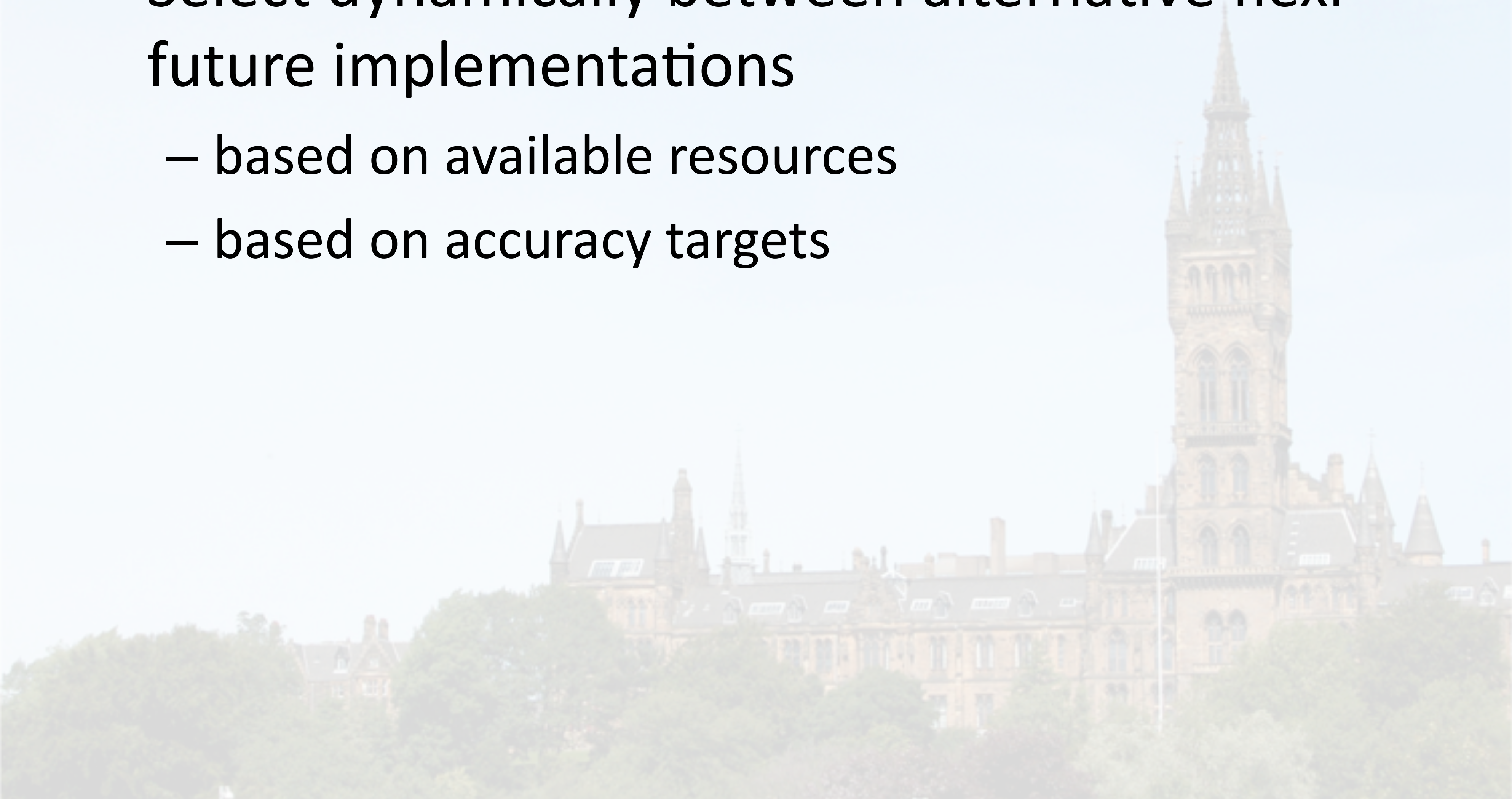


Compiler support

- Generate flexi-future alternatives
  - based on aggressive, non-conservative optimizations

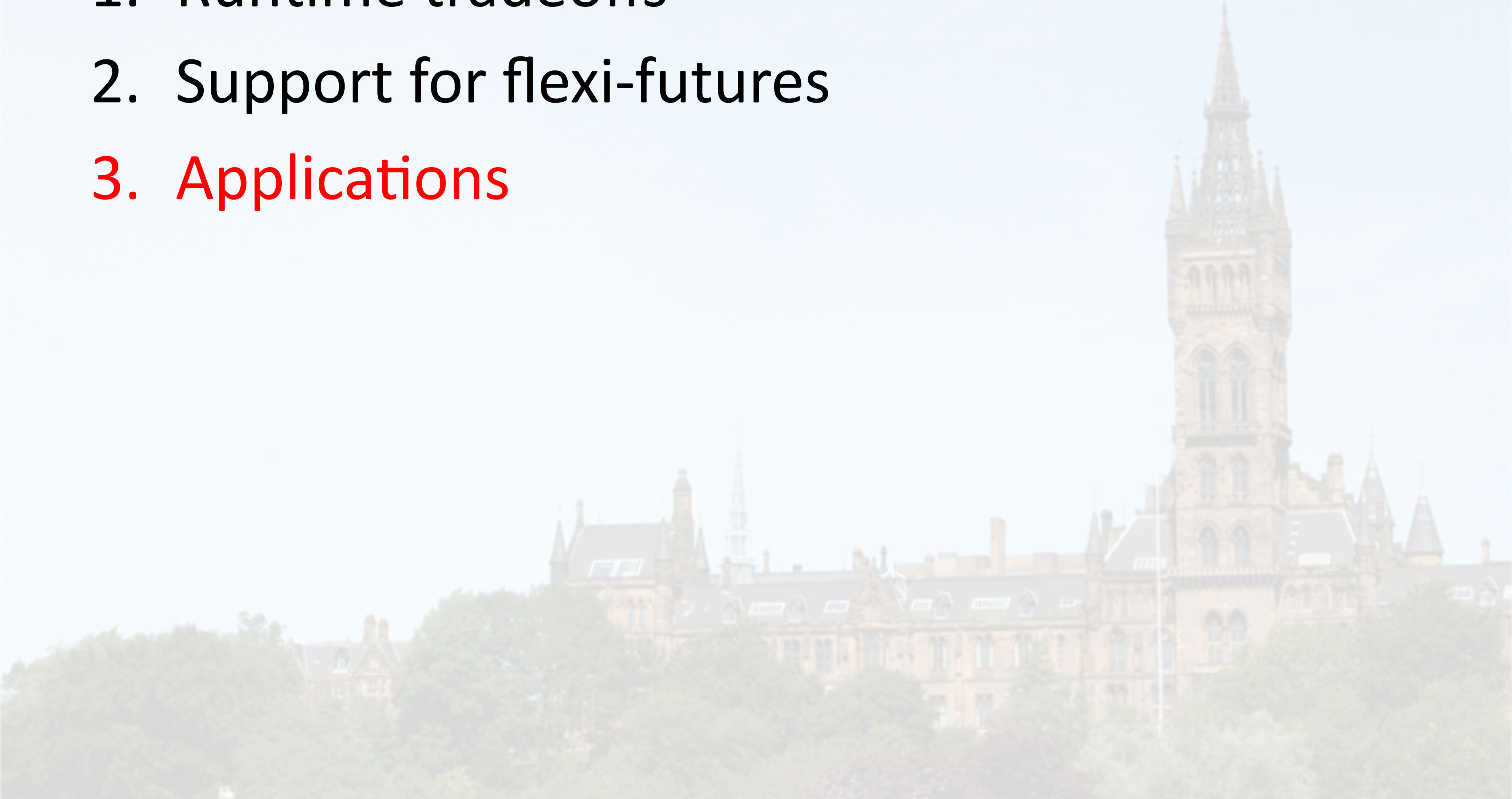
Runtime support

- Select dynamically between alternative flexi-future implementations
  - based on available resources
  - based on accuracy targets



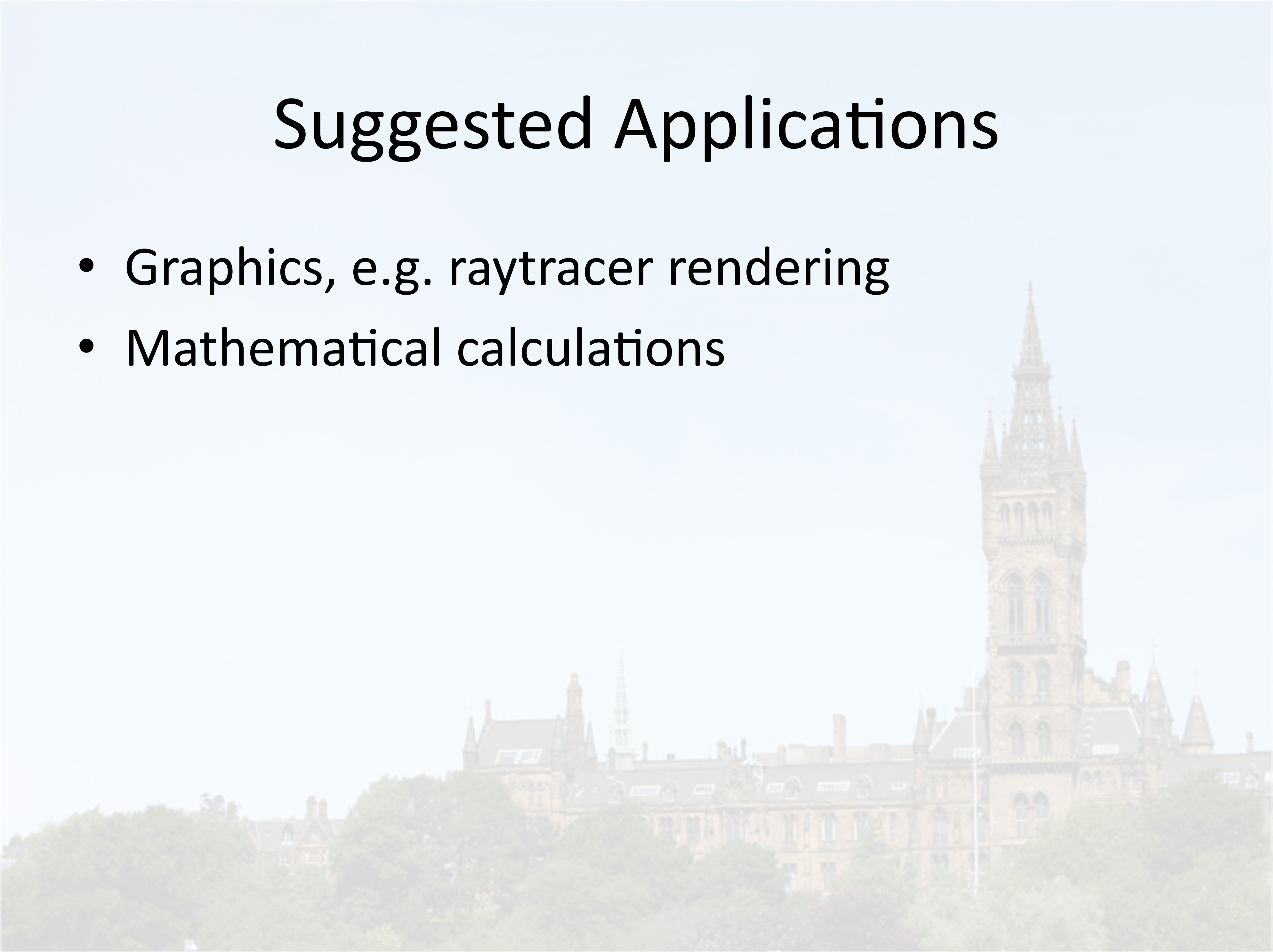
Outline

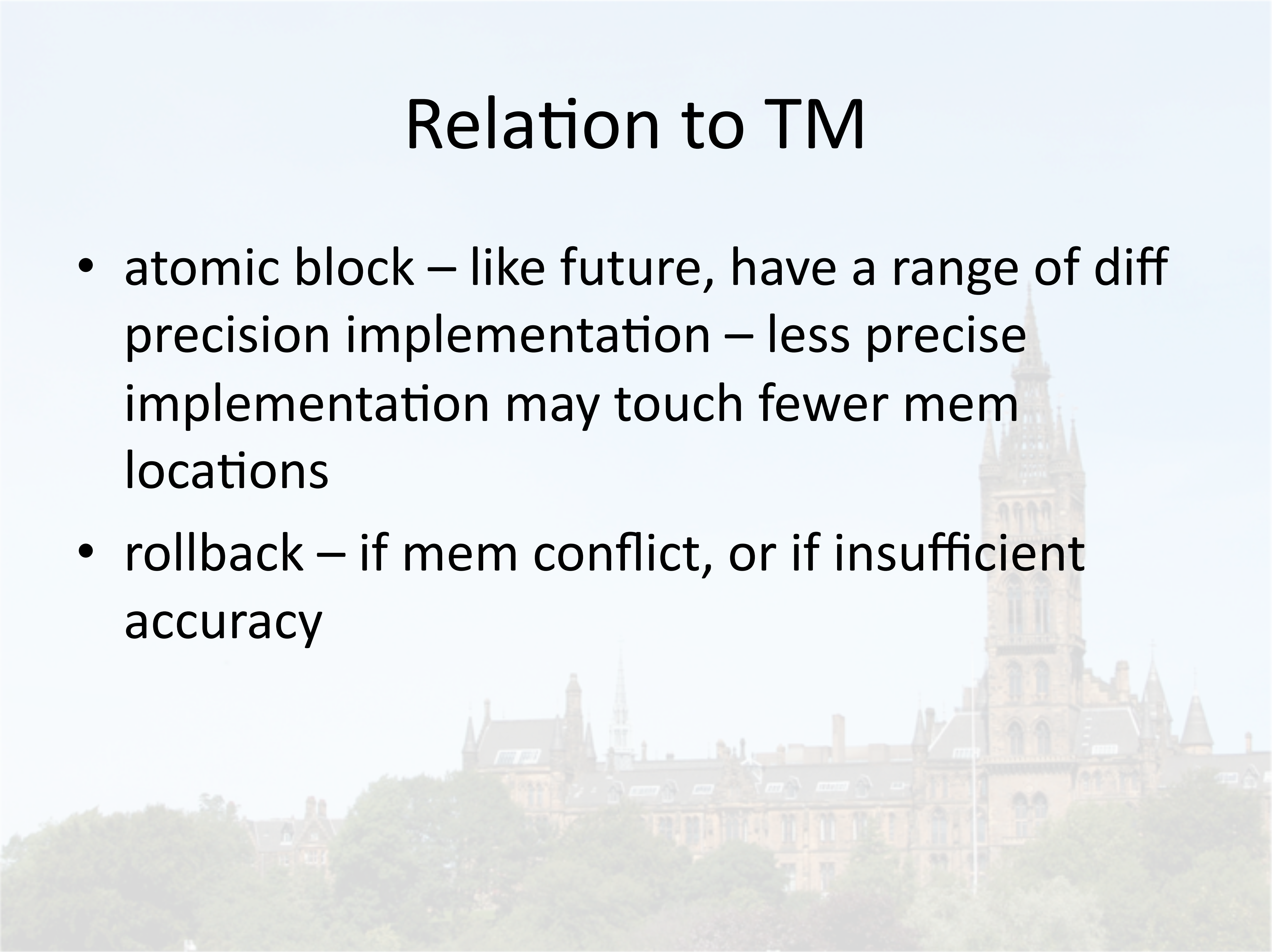
1. Runtime tradeoffs
2. Support for flexi-futures
3. Applications



Suggested Applications

- Graphics, e.g. raytracer rendering
- Mathematical calculations





Relation to TM

- atomic block – like future, have a range of diff precision implementation – less precise implementation may touch fewer mem locations
- rollback – if mem conflict, or if insufficient accuracy

Related Work

- Other projects/people along similar lines
  - MIT – Petabricks - multiple implementations of multiple algorithms to solve a problem - http://projects.csail.mit.edu/petabricks/
  - Lugano –ParaBoost - Virtual-Machine-Level Multi-Variant Speculation - http://sape.inf.usi.ch/paraboost
  - …

To do

- write up ideas
- vapour-ware -> prototype system
- vapour-paper -> real evalation