```
#include <amp_math.h>
void cenergy_2(      float * energygrid, extent<3> grid,
                float gridspacing, float z, int k,
                const float * atoms, int numatoms) {
    array_view<float,3> energygrid_view(grid, energygrid);
    array_view<float,2> energy_slice = energygrid_view(k);
    energy_slice.discard_data();
    array_view<const float,2> atom_view(numatoms,4,atoms);
    parallel_for_each(energy_slice.extent, [=](index<2> ji)
    restrict(amp) {
          float y = gridspacing * float(ji[0]);
          float x = gridspacing * float(ji[1]);
          float energy = 0.0f;
          for(int n =0; n < numatoms; n++) {
               float dx = x - atom_view(n,0);
               float dy = y - atom_view(n,1);
               float dz = z - atom_view(n,2);
               energy + = atom_view(n,3)/
                          precise_math::sqrtf(dx*dx + dy*dy+dz*dz);
          }
          energy_slice[ji] = energy;
    });
    energy_slice.synchronize();
}
```