Output format

The particle distributions are written to netCDF files.
===Litt diskusjon om ulik tankegang for partikkelfordelinger
Different formats may be available in the future (in particular ROMS float format)

LADIM format

This is a modfication of the LADIM 2 format. It is not backwards compatible, but scripts should be easy to modify. The change is done to follow the CF-standard as far as possible, and to increase flexibility.

The format is close to the format of "Indexed ragged array representation of trajectories" as described in appendix H.4.4 in the CF-documentation version 1.6. http://cf-pcmdi.llnl.gov/documents/cf-conventions/1.6/cf-conventions.html#idp8399648 There are however modifications, as we are more interested in the distribution of the particles at a fixed time than to follow the individual trajectories.

The dimensions are `time` and `particle_dim`. NetCDF 3 format allows only one unlimited dimension. As the duration and output frequency are known at the start of a ladim simulation, the particle dimension may be unlimited.

[===LARMOD. For larmod som kjøres fra symbioses-grensesnitt, larmod vet ikke nødvendigvis hvor lang simuleringen blir. Bruker derfor NetCDF-4 og begge dimensjoner ubegrenset]

The arrays `pstart(time)` and `pcount(time)` are used to adress the particle distributions. For a particle variable, for instance `X(particle_dim)`, the values at time `t` is found as (python index notation):

```
X[pstart[t] : pstart[t]+pcount[t]]
```

Particle identifier

The particle identifier, `pid` should always be present in the output file. It is a particle number, starting with 1 and increasing as the particles are released. The `pid` follows the particle and is not reused if the particle becomes inactive. In particular, `max(pid)` is the total number of particles involved in the simulation and may be larger than the number of active particles at a given time. It also has the property that if a particle is released before another particle, it has lower `pid`.

The particles identifiers at a given time frame *n* can be found from the output file as `pid_n = pid[pstart[n]:pstart[n]+pcount[n]]`. It has the following properties:

```
- pid_n is a sorted integer array
- pid_n[p] = pid[pstart[n]+p] >= p+1 with equality
    if all earlier particles are active at time frame n.
```

Example CDL

```
netcdf larmod_out {
dimensions:
      particle_dim = UNLIMITED ; // (868 currently)
      time = UNLIMITED ; // (217 currently)
variables:
      double time(time) ;
              time:long_name = "time" ;
              time:units = "seconds since 1970-01-01 00:00:00" ;
      int pstart(time) ;
              pstart:long_name = "start index for particle distribution" ;
      int pcount(time) ;
              pcount:long_name = "number of particles" ;
      int pid(particle_dim) ;
              pid:long_name = "particle identifier" ;
      float lon(particle_dim) ;
              lon:long_name = "longitude" ;
              lon:units = "degrees_east" ;
      float lat(particle_dim) ;
              lat:long_name = "latitude" ;
              lat:units = "degrees_north" ;

// global attributes:
              :history = "2013-05-10: created by LARMOD" ;
}
```