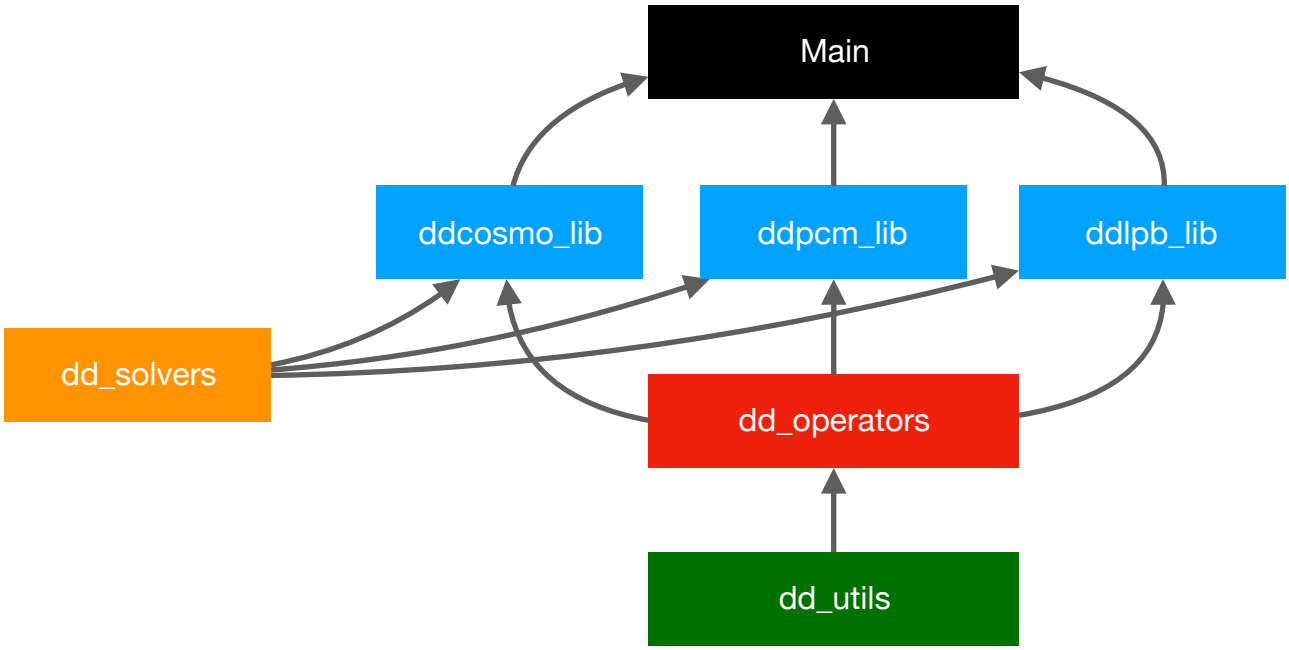
ddX

Modules:

- ddcosmo_lib
- ddpcm_lib
- ddlpb_lib
- dd_operators
- dd_solvers
- dd_utlis



Main

Filen ame: Main.f90

Use: ddcosmo_lib, ddpcm_lib, ddlpb_lib

Variables:
- phi, gradphi, hessphi (in)
- psi, gradpsi (in)
- do_force, do_fock (in)
- psimunu, phimunu (in, if do_fock)
- esolv (out)
- fx (out, if do_force)
- focksolv (out, if do_fock)

Subroutines:
- mkrhs
- readfile (new)

Content:
- call readfile (we only provide an example)
- ddinit
- call ddmain
- mkrhs: copy-paste from existing
- paraminit: read parameter file and initialise utility variables

Comments:
- The general idea is that this is a minimal (but sufficient) interface to couple ddX with an external driver routine (QM-code, Tinker, …)
- We kept the existing structure with a few exceptions:
    - The implementation of the subroutines **ddinit** and **ddfree** (former memfree) are moved dd_utils. This is the minimal interface with an enduser.

Interface to ddx (x=cosmo, pcm or lpb):
- In variables: phi, gradphi, hessphi, psi, gradpsi, psimunu, phimunu, do_force, do_fock
- Out variables: esolv, fx, focksolv

Interface to ddinit:
- In variables (mandatory): nsph, csph, rsph, epsout
- In variables (optional): epsin, kappa, eta, tol, lmax, ngrid, se

Module name: ddcosmo_lib

File name: ddcosmo.f90

Use: dd_solvers, dd_operators, dd_utils

Variables:
- X
- S
- g

Subroutines:
- ddcosmo: solve primal, compute energy, compute adjoint and force resp. Fock matrix if required:
    - call g = ddproject(-Phi): get rhs for primal linear system
    - call X = solver(Lx,g): solve primal linear system
    - esolv = 0.5*f(epsout)*sprod(psi,X)
    - if(do_fock or do_force)
        - call S = solver(Lstarx,psi)
    - endif
    - if(do_fock)
        - call xi = ddeval(S, at exterior points only with U_i weights -> check if this is true)
        - Focksolv = 0.5*f(epsout)*(sprod(psimunu,X) + sprod(xi,phimunu))
    - endif
    - if(do_force)
        - … (too complicated to be reported here!)
    - endif

Module name: ddpcm_lib

File name: ddpcm.f90

Use: dd_solvers, dd_operators, dd_utils

Variables:
- X, Phie
- S, Y, Q
- g (or Phi)

Subroutines:
- ddpcm: solve primal, compute energy, compute adjoint and force resp. Fock matrix if required:
    - call g = ddproject(Phi): get rhs for first primal linear system
    - call rhs = Rinfx(g): prepare rhs
    - call Phie = solver(Repsx,rhs): solve first primal linear system
    - call X = solver(Lx,-Phie): solve second primal linear system
    - esolv = 0.5*sprod(psi,X)
    - if(do_fock or do_force)
        - call S = solver(Lstarx,psi)
        - call Y = solver(Repsstarx, S)
        - Q = S-4*pi*frac{1}{epsout - 1}*Y
    - endif
    - if(do_fock)
        - call xi = ddeval(Q, at exterior points only with U_i weights -> check if this is true)
        - Focksolv = 0.5*(sprod(psimunu,X) + sprod(xi,phimunu))
    - endif
    - if(do_force)
        - … (too complicated to be reported here!)
    - endif

Module name: dd_operators

File name: dd_operators.f90

Use: dd_utils

Variables:

Subroutines:
- Lx, Lstarx, Ldm1x: Cosmo matrix, adjoint matrix, and diagonal preconditioner mat-vec operations
- D, Dstarx: global double layer operator and adjoint mat-vec
- Repsx, Repsstarx:
- Lkappax: HSP-mat-vec product with dd-strategy
- Sx, Sstarx: global single layer operator mat-vec
- Skappax, Skappastarx: global single layer operator mat-vec for LPB
- DtNx: local DtN mat-vec
- DtNkappax: local DtN mat-vec for LPB
- g (assemble rhs with U_i weight)
- gradL (former fdoka + fdokb)
- gradg (former fdoga)
- gradR
- gradS, gradSkappa, gradLkappa

Module name: dd_solvers

File name: dd_solvers.f90

Use: none

Variables:
- todo

Subroutines:
- gmres: main argument: matvec (specified in ddcosmo_lib, ddpcm_lib, ddlpb_lib)
- diis: main arguments: preconx, matvec (specified in ddcosmo_lib, ddpcm_lib, ddlpb_lib)

Module name: dd_utils

File name: dd_utils.f90

Use: none

Variables:
- model
- csph
- rsph
- nsph
- epsin, epsout
- kappa
- eta
- tol
- lmax
- ngrid
- shift (se)

Subroutines:
- ddinit
- ddfree (former memfree)
- sprod, fsw, dfsw

- ptcart, prtsph
- ylmbas, dbasis, polleg, trgev
- wghpot
- hsnorm, hnorm
- header
- calcv (required in Lx)
- adjrhs (required in Lstarx)
- intlmp (required in …)
- intrhs -> will be replaced
- ddmkxi (required in …) -> will be replaced
- ddproject (all, only interior, only exterior: from nodal to modal) -> replacement for intrhs
- ddeval (evaluation SH's series at integration points, + multiply optionally by U_i) -> replacement for ddmkxi