Lattice Tool Kit in Fortran90 (LTKf90)

http://nio-mon.riise.hiroshima-u.ac.jp/LTK/

Motivation and Objective

- Lattice is a tool for studying the Quantum Field Theories.
  - It is especcially useful for the hadron physics.
- Provide a readable code of essential parts of lattice QCD
  - Using it, non-specialist can make a code for his/her own purpose
  - Fortran90 which has “Module” can describe Data Structure as well as C++
  - Not a library, but a Tool kit, i.e., you construct a program for your purpose using any parts of LTKf90, and modifying them if necessary.

The program includes

- Metropolis and Pseudo-Heatbath gauge update
  - Standard Wilson Action
  - (1x1+1x2) Imroved Action
  - Anisotropic Lattice
- Hybrid Monte Carlo for dynamical fermions (Wilson Fermions)
- Quark Propagators
  - Wilson Fermion + Clover
- Parallelize with MPI
- Whole code is white box

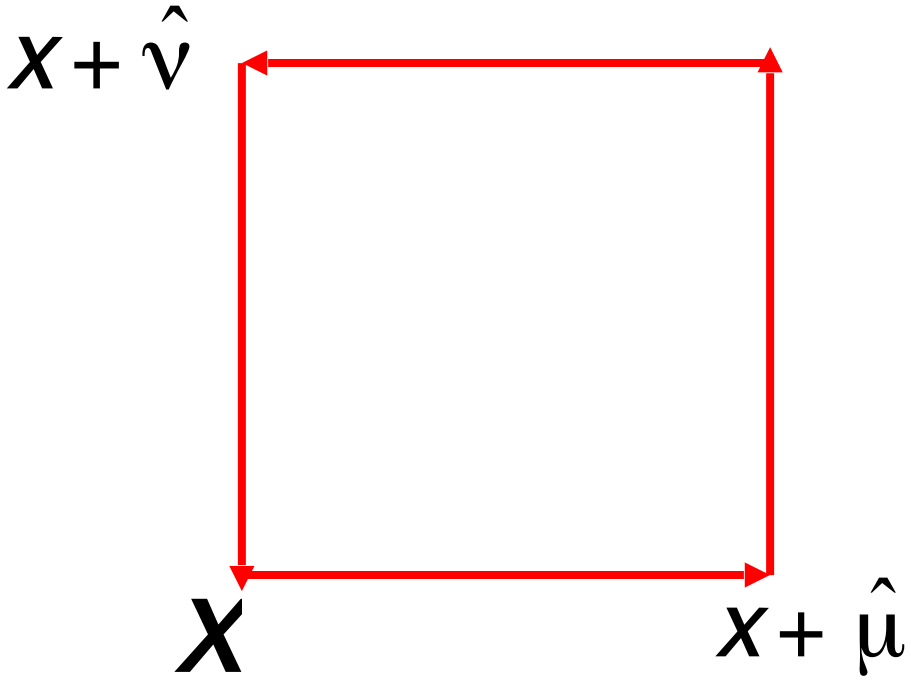
Module:Data structure+operand

```
c--------------------------c
      program ex1
c--------------------------c
      use vec_def
      TYPE(vector) p, q, r
      p = vector(0.5, -1.5); q = vector(0.5, 0.5)
      r = p + q
      write(*,*) r
      end
```

```
MODULE   v e c_def

        TYPE  v e c t o r
            r e a l   x, y
        END TYPE

        INTERFACE OPERATOR(+)
           MODULE PROCEDURE vadd
        END INTERFACE

        CONTAINS

        FUNCTION vadd(a,b) RESULT(c)
           TYPE(vector), INTENT(IN):: a, b
           TYPE(vector) c
             c %x = a%x + b%x
             c %y = a%y + b%y
        END FUNCTION

END MODULE
```

Ex. QCD(1) Plaquette

$$\sum_{x,\mu} U_\mu(x) U_\nu(x+\hat{\mu}) U_\mu^{+}(x+\hat{\nu}) U_\nu^{+}(x)$$



U_μ : SU(3)Matrix

TYPE(g_field1) staple, temp1, temp2, temp3

- t emp1 = u(nu)
- t emp2 = nu.gshift.u(mu)
- t emp3 = temp1 * temp2
- t emp1 = mu.gshift.u(nu)
- staple = staple + (temp3.prod1.temp1)

```
parameter( NC=3 )


TYPE g_field0
    SEQUENCE
    COMPLEX*16, DIMENSION
                    (NC,NC,-NDW+1:NX+NDW,-NDW+1:NY+NDW,
 &                              -NDW+1:NZ+NDW,-NDW+1:NT+NDW) :: g
    INTEGER parity, direction

 END TYPE


  TYPE g_field1
     SEQUENCE
     COMPLEX*16,  DIMENSION(NC,NC,NV/NBUSH) :: g
     INTEGER parity, direction
   END TYPE
```

```
c-------------------------------------------------------------------c
      FUNCTION  gadd(a,b) RESULT(c)
c-------------------------------------------------------------------c
      TYPE(g_field1), INTENT(IN):: a, b
      TYPE(g_field1) c

      do i = 1, NV/NBUSH

        c %g( 1, 1, i) = a%g( 1, 1, i) + b%g( 1, 1, i)
        c %g( 1, 2, i) = a%g( 1, 2, i) + b%g( 1, 2, i)
        c %g( 1, 3, i) = a%g( 1, 3, i) + b%g( 1, 3, i)
        c %g( 2, 1, i) = a%g( 2, 1, i) + b%g( 2, 1, i)
        c %g( 2, 2, i) = a%g( 2, 2, i) + b%g( 2, 2, i)
        c %g( 2, 3, i) = a%g( 2, 3, i) + b%g( 2, 3, i)
        c %g( 3, 1, i) = a%g( 3, 1, i) + b%g( 3, 1, i)
        c %g( 3, 2, i) = a%g( 3, 2, i) + b%g( 3, 2, i)
        c %g( 3, 3, i) = a%g( 3, 3, i) + b%g( 3, 3, i)

      enddo

      END FUNCTION
```

Ex. QCD(2)Quark propagators

$$S_{quark} = \bar{\psi}\Delta\psi$$

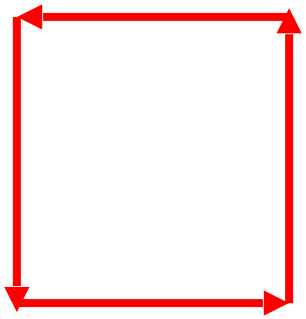
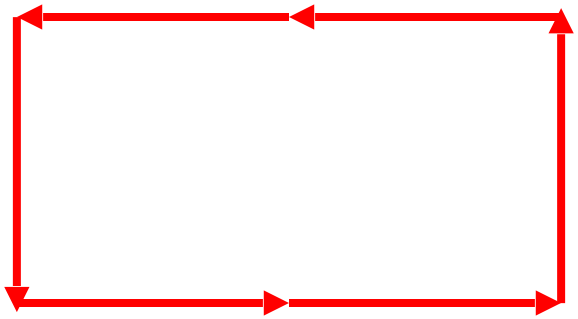
Δ^{-1} :Quark propagators

$$\Delta\vec{X} = \vec{B}$$

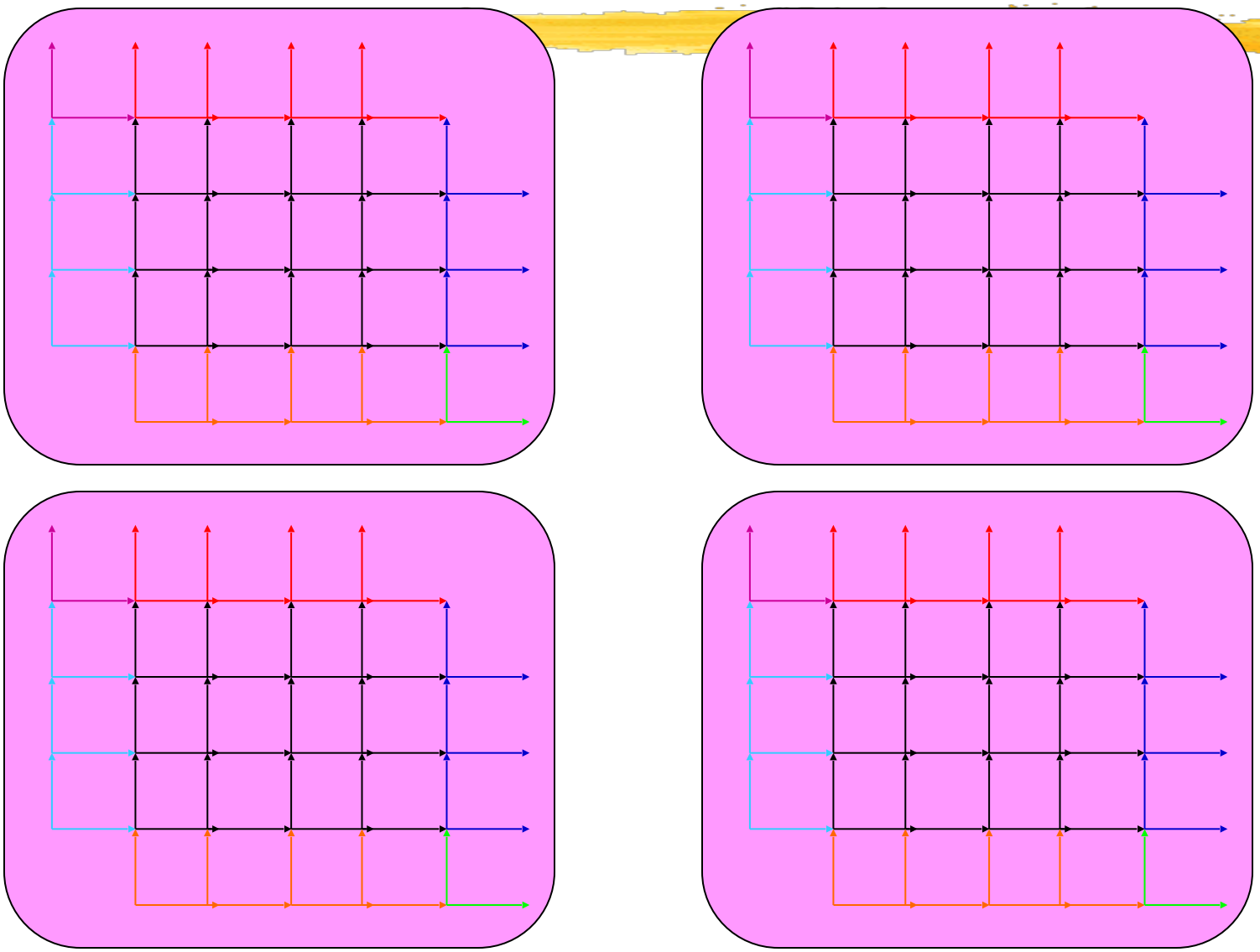
通常CG(共役勾配法)で解かれる

```
      TYPE(f_field) wxvect, x, b, res, p, q, s
      REAL*8 alpha, beta, c1, c2, c3

c     the initial condition  ( for i=0 )
      ! ...  res = b - W*x
      r es = b - wxvect(x,2)
      ! ...  p = W_adj * res
      p = wxvect(res,3)
c
      ! ...  c1 = < p | p >
      c 1 = p * p

      ! ...  the iteration starts
      do i = 1, imax
        ! ...  q = W * p
        q = wxvect(p,2)
        ! ...  c2 = < q | q >
        c 2 = q * q
```

Improved lattice actions

C0  + C1 

Parallelization: Add extra variables around each lattice.



Enjoy your own lattice QCD simulations !

Thank you !

감사합니다 !