
PARALLEL CODE: FOURIER TRANSFORM

e.g. FORWARD TRANSFORM

```
DO my_z=my_z%min , my_z%max
```

local fft over x:

```
(x,y,z) -> (kx_y,z)
```

```
real -> complex
```

```
real_field(x,y,my_z) -> complex_field(nx/2+1,ny,my_z)
```

local transpose (including x-dealias):

```
(kx,y,z) -> (y,kx,z)
```

```
complex_field(nx/2+1,ny,my_z) -> complex_field(ny,nx_dealias/2+1,my_z)
```

local fft over y:

```
(y,kx,z) -> (ky,kx,z)
```

```
(complex -> complex)
```

```
complex_field(ny,nx_dealias/2+1,my_z) -> complex_field(ny,nx_dealias/2+1,my_z)
```

y-dealias and re-pack:

```
complex_field(my_z,ny_dealias,nx_dealias/2+1)
```

```
END DO
```

Distributed transpose: I have all kxy for certain z. Send right kxy's to the right processor

```
DO p=1,N_p
```

```
SEND : Processor p gets sent
```

```
complex_field(all_my_nz,p*nxy_dealias/N_p:(p+1)*nxy_dealias/N_p - 1)
```

```
END DO
```

```
BARRIER
```

```
DO p=1,N_p
```

```
RECEIVE : Processor p receives
```

```
complex_field(1+nz*p/N_p:nz*(p+1)/N_p,all_the_kxy)
```

```
END DO
```