In this short note will be described the modifications needed in `hoppet` package in order to evolve (leading order only, for now) chiral–odd Fragmentation Functions. I started from `hoppet-1.1.2`.

1 File `dglap_choices.f90`

Just a declaration added:

```
integer, parameter, public ::  factscheme_FragCOddMSbar = 5
```

2 File `dglap_holders.f90`

In function `InitDglapHolder` added a `case (factscheme_FragCOddMSbar)` statement. Inside, the subroutine (`InitSplitMatFragCOddLO`) needed to initialize the splitting matrix for the chiral–odd evolution is called.

3 File `dglap_objects.f90`

Here is where the subroutine `InitSplitMatFragCOddLO` is defined. I just copied `InitSplitMatLO` and replaced `sf_Pgg`. . . functions with the new ones, `codd_Pgg`, etcetera.

4 File `splitting_functions.f90`

Four new functions are defined: `codd_Pgg`, `codd_Pqq`, `codd_Pgq` and `codd_Pqg`. `codd_Pqq` is the only one returning a non–zero value.

5 Check

The program has been tested against the “Kumano” program. I took a fake u–distribution ($x(1-x)$) and evolved it, in both programs, from $Q_0 = 70$ GeV to $Q = 100$ GeV, with LO evolution. A special care has been devoted to match α_s, since “Kumano” evolution equation is based on an explicit value for Λ_{QCD}. We checked that setting $\Lambda_{\mathrm{QCD}} = 0.2$ GeV, we obtain, to one loop,

$\alpha_s(M_Z) = 0.133861$ (with $M_Z = 91.187$ GeV), and so we put this values in the initializing section of the "Hoppet" program.

The results coincide with a precision better than 10^{-6}.