Appendix

A.1 How to install the program

The source code is available by `anonymous ftp` from `ftp.kek.jp` in the directory `kek/minami/grcnna`. The grc$\nu\nu\gamma$ system contains the complete set of FORTRAN sources for $e^+e^- \to \nu\bar{\nu}\gamma$ and $e^+e^- \to \nu\bar{\nu}\gamma\gamma$ matrix-elements and the four libraries, i.e., `BASES/SPRING`, `CHANEL`, `QEDPS` and utilities for kinematics. Those source codes are written in FORTRAN77. grc$\nu\nu\gamma$ has been developed on HP-UX, but should run on any UNIX platform with a FORTRAN complier.

The procedure of installation is as follows:

1. Editing *Makefile*.

   The following macros in *Makefile* should be taken care of by users themselves. For example, `GRCNNADIR` defines the directory name where grc$\nu\nu\gamma$ is installed. The values of `FC` and `FOPT` define the relevant compiler name and option for your system. The other macros can be left as they are.

`GRCNNADIR`	=	directory where grc$\nu\nu\gamma$ are installed.
`LIBDIR`	=	directory where libraries are installed. (default is `$(GRCNNADIR)/lib`.)
`BINDIR`	=	directory where an executable is installed. (default is `$(GRCNNADIR)/bin`.)
`MACHINE`	=	`[hpux|hiux|sgi|dec|sun]`
`FC`	=	FORTRAN compiler command name.
`FOPT`	=	FORTRAN compiler options.
`CERNLIBS`	=	CERNLIB including the jetset library. For HP-UX, `-L/cern/pro/lib -ljetset74 -lpacklib -lkernlib -L/lib/pa1.1/ -lm`

2. Compilation.

   By executing command `make install` the executable of the interface program is generated at `BINDIR`. Furthermore four libraries, `BASES/SPRING`, `CHANEL`, `QEDPS` and kinematics utility library, are generated in `LIBDIR`.

A.2 How to run the program

According to the number of hard-photons, two directories are prepared: `$(GRCNNADIR)/1a` and `$(GRCNNADIR)/2a`. Three FORTRAN files and a `Makefile` are stored in each directory.

For the numerical integration, `BASES`:

- `mainbs.f`: A main program of the numerical integration.
- `usrprm.f`: A subroutine for parameter setting.

For the event generation, `SPRING`:

- `mainsp.f`: A main program of the event generation. Event-loop is also included.

The subroutine, `usrprm.f`, is used commonly for both of `BASES` and `SPRING`.

Users can set the parameters of the calculations by editing the file `usrprm.f`. The key words are as follows;

```
* CM energy
      E0   = 170.d0
* Hard Photon condition
      Ea   =  1.d0
      THEa = 10.d0

* channel selection
*     Ineut=1   : sum up all neutrino species
*          =2   ; nu-e only
*          =3   : nu-mu (nu-tau )only
      Ineut = 1

* QED correction ********************************************
* QEDPS switch                                              !
*     Ips=0     : No ISR                                    !
*        =1     : ISR by QEDPS                              !
      Ips=1                                                 !
* QED correction ********************************************
```

```
* Higher order correction scheme ****************************
* Running alpha                                               !
*     Iruna=0   : On-shell scheme ; fixed alpha               !
*          =1   : On-shell scheme ; running alpha             !
*          =3   : Gmu scheme                                  !
      Iruna=1                                                 !
*                                                             !
* Running width                                               !
*     Irunw=0   : fixed width                                 !
*          =1   : running width                               !
      Irunw=1                                                 !
                                                              !
* Set QED alpha                                               !
      alpha0= 1.0d0/137.0359895d0                             !
      if(Iruna.ne.1) then                                     !
        alpha = 1.0d0/128.07d0                                !
       else                                                   !
        alpha = 1.0d0/137.0359895d0                           !
       end if                                                 !
* Higher order correction scheme ****************************

* WWgamma coupling
*
*     ang1a :     g1   =1 (SM)
*     andka : delta-k  =0 (SM)
*     anlma : lambda   =0 (SM)
      ang1a    =1
      andka    =0
      anlma    =0
```

The parameters for the anomalous TGC are available only for the single-photon case.

Users can proceed with the calculations as follows:

i) Change directory by typing

```
% cd $(GRCNNADIR)/1a
```

or

```
% cd $(GRCNNADIR)/2a
```

ii) Edit `usrprm.f` to set user parameters.

iii) Create an executable `integ` for the integration by typing

```
% make integ
```

iv) Numerical integration is actually performed by typing

```
%  integ
```

The results of the integration step are displayed on the console as well as written in an output file `bases.result`. The total cross section in *pb* and the estimated statistical error are shown on the last line, under `Cumulative Result`, in the table of the `Convergence Behavior for the Integration step`. The differential cross sections are also printed as a function of the energy, scattering angle of each particle and invariant masses of any two final particles. The probability distribution of the integrand is written in a file `bases.data` which will be used in the event generation step by `spring`.

v) Before running the event generation, users may edit `mainsp.f` to set additional parameters if needed and call the user's own analysis routines.

The following is the structure of the generated `mainsp.f`, where four-momentum of all particles including soft-photons generated by QESPS are stored in the `common/lujets/` in the `JETSET` format when subprogram `sp2lnd` is called in the event-loop:

```
Program  mainsp
implicit real*8(a-h,o-z)
external func
 .....................
real*4  p,v
```

```
      common/lujets/n,k(4000,5),p(4000,5),v(4000,5)
       .....................
       .....................
      mxtry  =   50
      mxevnt = 1000
      do 100 nevnt = 1, mxevnt

         call spring( func, mxtry )
       .....................
*        ------------------
          call sp2lnd
*        ------------------
*
*       ============================================
*       (  user_analysis based on the common lujets )
*       ============================================
*
  100 continue
       .....................
      stop
      end
```

iv) Create an executable `spring` for event generation by typing

```
% make spring
```

vii) Start the event generation by typing

```
% spring
```

Information concerning the event generation will be written in the `spring.result` file. Users should pay attention to the histograms generated in this step. The distributions of the generated events are superimposed with the character "0" on the histograms generated in the integration step. These two distributions should be consistent with each other within the statistical error of the generation. For the details of the output files of `BASES` and `SPRING`, users can consult Ref.[15].