

# Prospero2 说明文档

**作者:** 贾兴隆

Prospero2 是一个计算强子对撞机中 MSSM 超对称粒子(次)领头阶产生截面的 Fortran90 程序, 目前常见的 Fortran 编译器例如 gcc, g++, gfortran 都可以对其进行编译。

## Prospero2 官方网站

<https://www.thphys.uni-heidelberg.de/~plehn/index.php?show=prospero>

## 安装方法

下载后解压, 编译后即可使用。编译方法为在文件主目录内使用 `make` 指令。

## Prospero2 文件说明

该程序最重要的几个文件分别是

- 主程序文件 `prospero_main.f90`
- 可执行文件 `prospero_2.run`
- 输入文件 `prospero.in.les_houches`
- 输出文件 `prospero.dat`

如果需要立刻查看该程序包的运行方法, 可直接跳转至可执行文件 `prospero_2.run` 的介绍章节。

## 主程序文件 prospino\_main.f90

该文件包含了截面计算中最重要的某些参数设置，同时也是该程序包 main 函数所在的位置。每次修改该文件后都需要进行编译才能使更改生效。

这些参数依次是：

### inlo

该部分指定将要计算的是截面是领头阶 (LO) 的还是次领头阶 (NLO) 的。

如果 inlo = 0 则计算 LO 截面，如果 inlo = 1 则计算 NLO 截面。

```
!-----
  inlo = 1      ! specify LO only[0] or complete NLO (slower)[1]      !
!              ! results: LO      - LO, degenerate squarks, decoupling on !
!              !              NLO    - NLO, degenerate squarks, decoupling on !
!              !              LO_ms  - LO, free squark masses, decoupling off !
!              !              NLO_ms - NLO, free squark masses, scaled      !
!              ! all numerical errors (hopefully) better than 1%          !
!              ! follow Vegas iteration on screen to check                !
!-----
```

这个代码的注释中还提到了输出结果的含义，这一点本文档会稍后提及

### isq\_ng\_in

该部分指定 squark 质量是否是简并的。

如果 isq\_ng\_in = 0 则 squark 质量简并，如果 isq\_ng\_in = 1 则 squark 质量自由。如无特殊需求，该项应指定为 isq\_ng\_in = 1。

```

!-----
  isq_ng_in = 1      ! specify degenerate [0] or free [1] squark masses      !
                    ! [0] means Prospino2.0 with average squark masses      !
                    ! [0] invalidates isquark_in switch                    !
!-----

```

## icoll\_in & energy\_in

该部分指定强子对撞机的类型以及对撞能量。

如果 `icoll_in = 0` 则对撞机类型将指定为 `Tevatron` , 如果 `icoll_in = 1` 则对撞机类型将指定为 `LHC` 。

对撞能量 `energy_in` 需要手动指定, 单位是 `GeV` 。当前 `LHC` 的对撞能量为 `13 TeV` 即 `13000 GeV`

```

!-----
  icoll_in = 1      ! collider : tevatron[0], lhc[1]                        !
  energy_in = 13000 ! collider energy in GeV                             !
!-----

```

## i\_error\_in

该部分指定误差是否随能标改变。

如果 `i_error_in = 0` 则误差不随能标改变, 如果 `i_error_in = 1` 则随能标改变。如无特殊情况, 应设置为 `i_error_in = 0` 。

```

!-----
  i_error_in = 0    ! with central scale [0] or scale variation [1]        !
!-----

```

## final\_state\_in

该部分指定所计算截面的末态产生模式。

常用的有 `final_state_in = nn` 即 `neutralino/chargino` 对儿产生模式, 以及 `final_state_in = ll` 即 `slepton` 对儿产生模式。

```
!-----
final_state_in = 'nn'
!
!
!      ng      neutralino/chargino + gluino      !
!      ns      neutralino/chargino + squark      !
!      nn      neutralino/chargino pair combinations      !
!      ll      slepton pair combinations      !
!      sb      squark-antisquark      !
!      ss      squark-squark      !
!      tb      stop-antistop      !
!      bb      sbottom-antisbottom      !
!      gg      gluino pair      !
!      sg      squark + gluino      !
!      lq      leptoquark pairs (using stop1 mass)      !
!      le      leptoquark plus lepton (using stop1 mass)      !
!      hh      charged Higgs pairs (private code only!)      !
!      ht      charged Higgs with top (private code only!)      !
!
!      squark and antisquark added, but taking into account different sb or ss      !
!-----
```

## ipart1\_in & ipart2\_in

该部分用于指定所计算截面的末态粒子具体类型, 仅当末态产生模式被指定为 `ng`, `ns`, `nn`, `ll`, `tb`, `bb` 时生效。

以 `final_state_in = nn` 和 `final_state_in = ll` 两种模式为例:

当 `final_state_in = nn` 时, `ipart1_in` 和 `ipart2_in` 均指某一特定的 `neutralino/chargino`, 分别是

- 指定为 1, 2, 3, 4 时对应  $\tilde{\chi}_{1,2,3,4}^0$
- 指定为 5, 6 时对应  $\tilde{\chi}_{1,2}^+$
- 指定为 7, 8 时对应  $\tilde{\chi}_{1,2}^-$

当 `final_state_in = 11` 时, `ipart2_in` 此时不生效, 必须设置为 `ipart2_in = 1`, `ipart1_in` 对应特定的标轻子对儿产生, 例如 `ipart1_in = 1` 时指左手标电子对儿产生, `ipart1_in = 2` 时指右手标电子对儿产生。剩余产生类型可具体查看代码内的注释。

```

!-----
! ipart1_in = 7 !
! ipart2_in = 5 !
! !
! final_state_in = ng,ns,nn !
! ipart1_in = 1,2,3,4 neutralinos !
!           5,6 positive charge charginos !
!           7,8 negative charge charginos !
! ipart2_in the same !
! chargino+ and chargino- different processes !
! !
! final_state_in = ll !
! ipart1_in = 0 sel,sel + ser,ser (first generation) !
!           1 sel,sel !
!           2 ser,ser !
!           3 snel,snel !
!           4 sel+,snel !
!           5 sel-,snel !
!           6 stau1,stau1 !
!           7 stau2,stau2 !
!           8 stau1,stau2 !
!           9 sntau,sntau !
!          10 stau1+,sntau !
!          11 stau1-,sntau !
!          12 stau2+,sntau !
!          13 stau2-,sntau !
!          14 H+,H- in Drell-Yan channel !
! !
! final_state_in = tb and bb !
! ipart1_in = 1 stop1/sbottom1 pairs !
!           2 stop2/sbottom2 pairs !
! !
! note: otherwise ipart1_in,ipart2_in have to set to one if not used !
! !
!-----

```

## isquark1\_in & isquark2\_in

该部分用于指定所计算截面的末态粒子具体类型，仅当末态产生模式涉及轻味 squark 时生效。

```
!-----
! isquark1_in = 0 !
! isquark2_in = 0 !
! !
! for LO with light-squark flavor in the final state !
! isquark1_in = -5,-4,-3,-2,-1,+1,+2,+3,+4,+5 !
! (bL cL sL dL uL uR dR sR cR bR) in CteQ ordering !
! isquark1_in = 0 sum over light-flavor squarks throughout !
! (the squark mass in the data files is then averaged) !
! !
! flavors in initial state: only light-flavor partons, no bottoms !
! bottom partons only for Higgs channels !
! !
! flavors in final state: light-flavor quarks summed over five flavors !
! !
!-----
```

## 其余事项

该文件的其它内容均为 main 函数的语句，但该文件中有一部分注释可能需要注意：

```
!-----
! input file: prospino.in.leshouches !
! use block MASS for masses, plus low-energy mixing matrices !
! !
! output file: prospino.dat for compact output format !
! prospino.dat2 for long output including subchannels !
! prospino.dat3 lo file for masses, flags, etc !
!-----
```

该注释告诉用户，该程序的输入文件为 `prospino.in.leshouches`，输出文件为 `prospino.dat`, `prospino.dat2`, `prospino.dat3`。文档后面的章节会详细介绍这些内容。

## 可执行文件 `prospino_2.run`

每当 `prospino_main.f90` 中的配置信息发生修改时，都需要使用 `make` 命令进行编译才能使更改在可执行文件 `prospino_2.run` 中生效。

二进制文件的使用方法为直接执行，当位于程序主目录时，执行命令为 `./prospino_2.run`。

**需要注意的是，如果想在其他位置运行 `prospino_2.run`，则需要一些额外的准备工作。这一点在文档末尾的特殊情况中会进行介绍。**

直接运行时会有很多信息打印在屏幕上，如果不希望这些信息被打印出来，则可以使用诸如 `./prospino_2.run > a.txt` 这样的命令，把输出信息存放在一个名为 `a.txt` 的文件中。

## 输入文件 `prospino.in.les_houches`

这是一个标准的 SLHA 格式文件，其中包含的主要内容有**粒子质量谱，混合矩阵**等信息。

运行 `prospino_2.run` 后 `Prospino2` 程序就从这个文件中读取信息。`prospino_main.f90` 中指定好具体的末态粒子类型后，程序也是从 `prospino.in.les_houches` 中根据相应粒子的粒子编号获取质量信息。

## 输出文件

当截面计算结束后，计算的结果会存放在下面的输出文件中，这些文件会输出在执行 `prospino_2.run` 时终端所处的位置。也就是说，如果在程序主目录中执行 `prospino_2.run` 则输出文件就会输出在程序主目录中。

### `prospino.dat`

运行 `prospino_2.run` 后，等待些许时间计算方可结束，此时结果就被存放在一个名为 `prospino.dat` 的文件中，这个文件的内容是如下一个类似表格的形式：



```

nn  5  7      0.0    0.0    1.0 107.5 107.5 0.000 5.28      0.761E-03 6.92      0.615E-03 1.3091 5.50      7.21

      i1 i2  dummy0 dummy1 scafac  m1      m2      angle L0[pb]  rel_error NLO[pb]  rel_error  K      L0_ms[pb] NLO_ms[pb]

```

第一行为数值结果，第二行为第一行的补充信息。

- 最开始的 `nn` 表示末态产生模式。
- `5, 7` 对应 `i1 i2` 表示具体的末态粒子类型。
- `scafac` 表示能标因子。
- `m1 m2` 表示输入文件中对应的 `nn` 模式 `5 7` 粒子的质量。
- `angle` 表示混合角，为 `0` 时没有混合。
- `L0[pb]` 表示考虑 `squark` 质量简并时的领头阶截面，单位为 `pb`。紧随其后的 `rel_error` 表示该数值的相对误差。
- `NLO[pb]` 表示考虑 `squark` 质量简并时的次领头阶截面，单位为 `pb`。紧随其后的 `rel_error` 表示该数值的相对误差。
- `K` 表示 `K` 因子，即次领头阶截面大小与领头阶截面大小的比值。
- `L0_ms[pb]` 表示**不考虑** `squark` 质量简并假设时的领头阶截面，单位为 `pb`。
- `NLO_ms[pb]` 表示**不考虑** `squark` 质量简并假设时的次领头阶截面，单位为 `pb`。
- 暂时不清楚 `dummy0 dummy1` 的具体含义。

需要注意的是，如无特殊需要，请使用 `L0_ms[pb]` 和 `NLO_ms[pb]` 作为输出数值结果。

## prospino.dat2 和 prospino.dat3

这两个文件也是输出文件，相比 `prospino.dat` 它们包含了更多的信息。`prospino.dat2` 包含了所有的质量信息，`prospino.dat3` 则包括了几乎所有输入参数和输入文件信息。

## 其余文件

除了上述文件之外，还有一些对 `Prospino2` 十分重要，但是绝大多数情况下不需要修改的文件。

## Xvital.f90

该文件存放着标准模型的输入参数，绝大多数情况下不需要修改。

## Xprospino\_subroutine.f90

该程序用于解析输入文件，因此该文件与输入文件的具体数据格式有关，在使用其提供的标准文件的情况下，该文件也不需要修改。

# Prospino2 的某些特殊情况

Prospino2 是一个只能计算 MSSM 模型粒子产生截面的程序，也就是说它只支持 4 个 neutralinos 的产生，而超对称的大多数衍生模型并不是这种最简情形。同时 Prospino2 在 prospino\_main.f90 中的程序设置上显示只能设置第一代和第三代标轻子产生过程作为计算截面。

## neutralinos 部分

以 NMSSM 模型为例，由于有 siglino 的存在所以一共存在 5 个 neutralinos，此时如果想使用 Prospino2 计算截面则需要将 NMSSM 的 neutralinos 部分退化为 MSSM 的情况，这要求 Electroweakinos 之间的混合效应不大。在满足混合不大的前提下，具体的操作方法为，舍去 siglino 以及 siglino 为主要成分的 neutralino。举例来说，如果  $\tilde{\chi}_1^0$  为 siglino 则混合矩阵需要去掉第一行以及第五列，这样就可以从 5x5 的矩阵退化为 4x4 的矩阵，如果  $\tilde{\chi}_2^0$  为 siglino 则需要去掉第二行以及第五列。

## Smuon 相关截面的计算

在 prospino\_main.f90 中的 ipart1\_in & ipart2\_in 部分可以看到，末态粒子类型只能选择第一代和第三代标轻子，而没有第二代标轻子 Smuon 的选项。由于 Smuon 和 Selectron 所参与的相互作用完全一样，所以这个问题的解决方法是，在 prospino.in.les\_houches 中将 Smuon 的质量信息填入到 Selectron 对应的位置。此时如果选择计算 Selectron 相关的产生截面，那么实际上计算的其实是 Smuon 相关的产生截面。

例如，现在假设 Selectron 和 Smuon 的质量分别为 100 GeV 和 200 GeV。此时如果要计算左手 Selectron 对儿产生截面，则在 prospino.in.les\_houches 文件中的 1000011 位置填入 100 GeV。如果要计算左手 Smuon 对儿产生截面，则需要在 1000011 位置填入 200 GeV。

其余的诸如右手标轻子，标中微子信息也是同样的处理方法。

## 不位于程序主目录时如何运行 Prospero

对于很多程序的可执行文件来说，与该文件保持目录同级都不是一个必须的要求，如果位于其他位置，输入相应的路径也可以直接执行。以 Prospero 为例，假设程序主目录位于一个名为 Prospero2 的文件夹中，当前所处位置为该目录的上级目录，则直接运行可执行文件的命令是：

```
./Prospero2/prospino_2.run
```

但如果真的这样直接运行，就会遇到报错，会提示某些子程序和 `prospino.in.les_houches` 文件找不到。这说明运行位置的目录下至少要包含这些子程序和 `prospino.in.les_houches` 文件，而 `prospino_2.run` 所在的位置其实并不重要，这是因为编译后所有的程序信息都已经存在于该二进制文件中了。但要注意，编译操作仍然必须要在程序主目录中进行。

程序主目录中存在一个名为 `Pro2_subroutines` 的文件夹，该文件夹里存放了运行 `prospino_2.run` 时所需的全部子程序。该文件夹下的程序不会因为 `prospino_main.f90` 的变化而发生变化。

因此，要想在其他位置运行 `prospino_2.run`，正确的做法是把 `Pro2_subroutines` 文件夹以及 `prospino.in.les_houches` 文件复制到要执行命令的目录。

例如，假设 `prospino_2.run` 位于任意路径 `PATH` 下，则运行命令为：

```
PATH/prospino_2.run
```

假设运行此命令时，终端位于 `RUN_PATH` 路径下，则 `RUN_PATH` 下必须存在子程序文件夹 `Pro2_subroutines` 以及输入文件 `prospino.in.les_houches` 才能运行成功，`prospino.dat` 等输出文件也会输出在 `RUN_PATH` 下。

这也使多进程运行同一个 Prospero 成为了可能，只需要在多个目录下存放 `Pro2_subroutines` 以及 `prospino.in.les_houches` 并分别在这些目录下执行同一个 `prospino_2.run` 即可。