

Orbit Fitting Library - IDL

The IDL `orbit_lib` package was written by Gail Schaefer to fit binary orbits based on the following types of data sets:

- Visual binary orbit.
- Spectroscopic binary orbit (single-lined or double-lined).
- Simultaneous orbit fit to visual and spectroscopic data.

Download the IDL Binary Grid Search Software

Download the IDL `orbit_lib` package here (http://www.astro.gsu.edu/~schaefer/chara/software/orbit_lib.tar) (updated 2023Sep26). Unpack the tar file and include the `orbit_lib` directory in your IDL path.

Running the orbit fitting code requires the IDL astronomy library maintained by NASA Goddard Spaceflight Center and the Coyote Graphics library. These packages can be downloaded through the following links:

- IDL Astronomy Library (<https://idlastro.gsfc.nasa.gov/>)
- IDL Coyote Graphics Library (<http://www.idlcoyote.com/documents/programs.php>)

Visual Binary Orbits

These are the main routines available in `orbit_lib` for fitting visual binary orbits:

- `test_chi.pro`: Performs a grid search for orbital solutions by varying the period (P), time of periastron passage (T), and eccentricity (e). The program follows the three-dimensional grid search procedure described by Hartkopf et al. (1989, AJ, 98, 1014) (<https://ui.adsabs.harvard.edu/abs/1989AJ.....98.1014H/abstract>) and Mason et al. (1999, AJ, 117, 1023) (<https://ui.adsabs.harvard.edu/abs/1999AJ....117.1023M/abstract>). The routine will ask the user to input initial values, step sizes, and search ranges for P, T, e. The program will perform a grid search to find the orbital solution within the search ranges where the `chi2` is at the minimum.
- `hartkopf_mc.pro`: Randomly varies P,T,e within specified ranges to map the `chi2` surfaces rather than searching through a uniform grid as done above. It will ask for initial values and search ranges for P, T, e. The program will randomly select values for P, T, e within the search ranges and optimize the remaining orbital parameters for each iteration. It will continue until it finds a specified number of solutions within a given `chi2` interval (both specified by the user). The routine finds the best fit orbital solution with where the `chi2` is minimum and also saves a file called `result_chi` that contains all solutions retained within the specified `chi2` interval.

- `hartkopf_mc_mass.pro`: Same as `hartkopf_mc.pro` but only selects orbits that have total masses within a specified mass interval. This version of the code requires the user to enter a parallax for the system so that the total mass can be computed through Kepler's Third Law.
- `newt_raph.pro`: Performs a formal orbit fit using a Newton-Raphson method to minimize chi2 by calculating a first-order Taylor expansion for the equations of orbital motion. The user selects which orbital parameters to hold fixed and which to optimize. The user inputs initial starting parameters.
- `newt_raph_ell.pro`: Same as `newt_raph.pro` but uses error ellipses as input rather than uncertainties in separation and position angle.

The first three routines are useful for binaries with limited orbital coverage or for computing an initial estimate of the orbital parameters because they can search through a large range of orbital parameter space to map out the chi2 surfaces. The Newton Raphson routines can be used for well-defined orbits and require a reasonable initial estimate for the orbital parameters.

The format for the input data file for `test_chi.pro`, `hartkopf_mc.pro`, and `newt_raph.pro` is:

```
time PA err_PA sep err_sep
```

where time is the time of the observation, PA is the position angle (east of north) measure in degrees, `err_PA` is the error in position angle, `sep` is the binary separation, and `err_sep` is the error in the separation.

For `newt_raph_ell.pro`, the format for the input data file is:

```
time PA sep sigma_major sigma_minor phi
```

where the uncertainties in the binary position are given as an error ellipse defined by the semi-major axis (`sigma_major`), semi-minor axis (`sigma_minor`), and position angle of the major axis (`phi`).

The orbital parameters determined by the visual orbit fitting include:

- P: period (in same units as time)
- T: time of periastron passage (in same units as time)
- e: eccentricity
- a: semi-major axis (in same units as separation)
- i: inclination (in degrees)
- Ω : position angle of the line of nodes
- ω_B : argument of periastron for the companion ($\omega_B = \omega_A + 180^\circ$)

There is a 180° ambiguity in Ω and ω_B for visual orbits.

Spectroscopic Radial Velocity Orbits

There are three routines for fitting spectroscopic orbits, depending on if it is a single-lined (SB1) or double-lined (SB2) spectroscopic binary, or if both SB1 and SB2 data exist.

- `fit_velocity_sb1.pro`: Fit a radial velocity curve for a single-lined spectroscopic binary orbit.
- `fit_velocity_sb2.pro`: Fit radial velocity curves for a double-lined spectroscopic binary orbit.
- `fit_velocity_sb2sb1.pro`: Fit radial velocity curves when both SB1 and SB2 data exist.

Each of these routines requires the user to enter reasonable estimates for the initial orbital parameters. The user can select which parameters to hold fixed and which to optimize.

The format for the input SB1 data file is:

```
time V1 err_V1
```

where time is the time of observation, V1 is the radial velocity of the primary, and err_V1 is the error in the radial velocity.

The format for the input SB2 files is:

```
time V1 err_V1 V2 err_V2
```

where time is the time of observation, V1 is the radial velocity of the primary, err_V1 is the error in the primary radial velocity, V2 is the radial velocity of the secondary, and err_V2 is the error in the secondary radial velocity.

The orbital parameters determined by the spectroscopic orbit fitting include:

- P: period (in same units as time)
- T: time of periastron passage (in same units as time)
- e: eccentricity
- K1: radial velocity amplitude of the primary
- K2: radial velocity amplitude of the secondary (for SB2 orbits only)
- ω_A : argument of periastron for the primary ($\omega_A = \omega_B + 180^\circ$)
- Vsys: systemic velocity for the system

Simultaneous Fit to Visual and Spectroscopic Orbits

These are the main routines for fitting orbits simultaneously to visual and spectroscopic binary data:

- fit_orbit_vbsb2.pro: Compute simultaneous orbit for visual and SB2 binary data. Uncertainties for visual binary are given as err_PA and err_sep.
- fit_orbit_vbsb2_ell.pro: Compute simultaneous orbit for visual and SB2 binary data. Uncertainties for visual binary data are given as error ellipse (sigma_major, sigma_minor, phi).
- fit_orbit_vbsb1.pro: Compute simultaneous orbit for visual and SB1 binary data. Uncertainties for visual binary are given as err_PA and err_sep.
- fit_orbit_vbsb1_ell.pro: Compute simultaneous orbit for visual and SB1 binary data. Uncertainties for visual binary data are given as error ellipse (sigma_major, sigma_minor, phi).

The data formats for each set of data are the same as given in the sections on Visual Binary Orbits and Spectroscopic Binary Orbits. Each of the routines performs a formal orbit fit using a Newton-Raphson method to minimize chi2 by calculating a first-order Taylor expansion for the equations of orbital motion. The user selects which orbital parameters to hold fix and which to optimize.

The orbital parameters determined through the simultaneous visual and spectroscopic orbit fitting include:

- P: period (in same units as time)
- T: time of periastron passage (in same units as time)
- e: eccentricity
- a: semi-major axis (in same units as separation)

- i : inclination (in degrees)
- Ω : position angle of the line of nodes
- ω_A : argument of periastron for the primary ($\omega_A = \omega_B + 180^\circ$)
- K_1 : radial velocity amplitude of the primary
- K_2 : radial velocity amplitude of the secondary (for SB2 orbits only)
- V_{sys} : systemic velocity for the system

Note that the simultaneous orbit fitting routines adopt the spectroscopic convention of defining the argument of periastron relative to the primary (ω_A) rather than the visual orbit convention of defining it relative to the secondary (ω_B).

For visual+SB2 orbits, the individual masses M_1 and M_2 and system distance (orbital parallax) can be computed from the orbital parameters. See equations in `compute_err.pro`.

When fitting a simultaneous orbit, it is often useful to fit each set of data (visual vs. spectroscopic) independently first to determine proper weighting. This is typically done by scaling the uncertainties for each set of data to force the reduced $\chi^2 = 1$ for the visual-only fit and the spectroscopic-only fit. The routines `scale_pasep.pro`, `scale_pasep_ell.pro`, `scale_velocity_sb1.pro`, and `scale_velocity_sb2.pro` can be used to scale the uncertainties based on the reduced χ^2 values from the individual fits.

Example 1: Visual Binary Orbit - χ^2 search

Course grid steps:

```
IDL> test_chi, 'sample_vb_data', 42
Enter P T e (period, time of periastron passage, eccentricity):
: 50 2000 0.5

This program will only search for solutions of the time of
periastron passage within +/- P/2 of the chosen epoch.
Enter epoch of periastron passage:
: 2000

Determine searching ranges and step sizes for P,T,e.
step is the incremental size for which to vary given parameter
min is the lower value for the searching range, ie:  $P_i = P - \text{minP}$ 
max is the higher value for the searching range, ie:  $P_f = P + \text{maxP}$ 

Enter Pstep size, minP, maxP for search in P:
: 1 20 20
Enter Tstep size, minT, maxT for search in T:
: 1 50 50
Enter estep size, mine, maxe for search in e:
: 0.1 0.5 0.9
Best fit P T e  $\chi^2$ 
39.000000 1982.5000 0.40000000 149.46931
```

Finer grid steps:

```

IDL> test_chi,'sample_vb_data',42
Enter P T e (period, time of periastron passage, eccentricity):
: 39.000000 1982.5000 0.40000000

This program will only search for solutions of the time of
periastron passage within +/- P/2 of the chosen epoch.
Enter epoch of periastron passage:
: 2000

Determine searching ranges and step sizes for P,T,e.
step is the incremental size for which to vary given parameter
min is the lower value for the searching range, ie:  $P_i = P - \min P$ 
max is the higher value for the searching range, ie:  $P_f = P + \max P$ 

Enter Pstep size, minP, maxP for search in P:
: 0.1 2 2
Enter Tstep size, minT, maxT for search in T:
: 0.1 2 2
Enter estep size, mine, maxe for search in e:
: 0.01 0.2 0.2
Best fit P T e  $\chi^2$ 
  40.600000 1981.8000 0.36000000 148.52482
reduced  $\chi^2$ : 1.9288938
A B F G: 13.345161 75.492536 89.897107 1.5114096
a i Omega omega: 93.252840 141.08653 20.896201 294.98878
P T e a i Omega omega
  40.600000 1981.8000 0.36000000 93.252840 141.08653 20.896201 294.98878
Errors determined from covariance matrix:
  3.4647461 1.7455763 0.089004305 1.2176848 4.3236015 7.6347056 8.0712485

```

Map out the χ^2 surfaces to look for correlations between the orbital parameters:

```

IDL> hartkopf_mc, 'sample_vb_data', 42
Enter P T e (period, time of periastron passage, eccentricity):
: 40.629674 1981.8552 0.35774869 76.948622

This program will only search for solutions of the time of
periastron passage within +/- P/2 of the chosen epoch.
Enter epoch of periastron passage:
: 2000

Determine searching ranges for P,T,e.
Search range goes from min to max:

Enter minP, maxP for search in P:
: 30 70
Enter minT, maxT for search in T:
: 1960 1990
Enter mine, maxe for search in e:
: 0.0 0.7
Unscaled chi2: 148.61150
Do you want to scale chi2 so that chi2red = 1.0 (y/n)?
: y
Scaled chi2: 77.000000
Scale factor for chi2: 0.51812948
Enter the variation of chi2 you wish to examine:
: 9
Enter number of solutions you wish to find within this variation of chi2:
: 10000
Enter seed for random generator (ex. 1001L):
: 1001L
Best fit P T e chi^2
  40.629674 1981.8552 0.35774869 76.948622
reduced chi2: 0.99933275
A B F G: 13.856918 75.671301 89.629740 0.82546064
a i Omega omega: 93.036248 141.46710 20.970178 295.47775
P T e a i Omega omega
  40.629674 1981.8552 0.35774869 93.036248 141.46710 20.970178 295.47775
Errors:
  2.4831132 1.2353693 0.063634132 0.86162402 3.1255553 5.5773605 5.9806698

```

In this example, the output file results_chi is saved to sample_chi_3sig. The range of orbital solutions and chi2 maps can be examined using plot_PaeM.pro:

Compile the code:

```

IDL> .r plot_PaeM

IDL> plot_PaeM,'sample_vb_data',42,'sample_chi_3sig',10000
Enter the parallax of the system (in same units as separation):
(distance of 140, parallax = 7.14 mas)
: 7.14
Is period in years or days (y/d)?
: y
Computing masses...
Error estimates: min max mean median stddev
M: 0.9371 2.5195 1.1806 1.0842 0.2558
P: 33.3811 58.6161 47.0460 47.3919 6.0123
a: 90.0490 107.3089 97.2443 96.5830 3.9494
e: 0.033690 0.583895 0.233071 0.212091 0.123668
i: 128.7618 151.9118 143.2954 143.7208 3.9938
T: 1970.0972 1985.3360 1978.6495 1978.7093 3.2056
W: 7.5378 54.7040 34.6229 36.2734 12.4452
w: 281.7177 350.0157 309.4223 308.3091 14.8131
Minimum chi2: 76.964248
Enter maximum mass for chi2 plots:
: 3
M and a are cut-off in chi2 plots
1-sigma Error estimates: min max mean median stddev
P: 37.93 44.73 41.33 41.37 1.65
T: 1979.74 1983.12 1981.49 1981.48 0.82
e: 0.2621 0.4325 0.3426 0.3377 0.0412
a: 92.12 94.45 93.36 93.36 0.58
i: 137.14 145.32 141.85 142.15 1.95
W: 15.22 30.73 22.58 22.42 3.68
w: 289.32 305.64 297.02 297.01 4.02
M: 1.14 1.61 1.31 1.29 0.11
1-sigma confidence intervals:
P: 41.078492 + 3.6499680 - 3.1464570
T: 1981.5594 + 1.5609000 - 1.8223000
e: 0.34679224 + 0.085693510 - 0.084649910
a: 93.218418 + 1.2324435 - 1.0969598
i: 141.65350 + 3.6705210 - 4.5182977
Omega: 22.178528 + 8.5525728 - 6.9545528
omega: 296.28289 + 9.3543604 - 6.9630751
Mtot: 1.3188080 + 0.29002460 - 0.17874300

```

For a reliable orbit, the 1-sigma chi2 errors will be the same as the formal errors determined from the newt_graph.pro code.

The files temp.ps and temp_orb.eps show cross-cuts through the chi2 surfaces for different orbital parameters. The confidence intervals for individual parameters are color coded:

Red: 1-sigma ($\Delta \chi^2 = 1$)

Blue: 2-sigma ($\Delta \chi^2 = 4$)

Green: 3-sigma ($\Delta \chi^2 = 9$)

Example 2: Visual Binary Orbit - formal orbit fit

Compute a formal visual binary fit:

```
IDL> newt_raph,'sample_vb_data',42
Enter P,T,e,a,i,Omega,omega:
: 40.6 1981.8 0.36 93.252840 141.08653 20.896201 294.98878
Vary each orbital element?
For each element, enter 0 to hold element fixed, 1 to vary:
[P,T,e,a,i,Omega,omega]
: 1 1 1 1 1 1 1
Final values:
Orbital Element Error
P: 41.385217 3.6817768
T: 1981.3805 1.8722253
e: 0.34021606 0.091073331
a: 93.334535 1.2271341
i: 141.85568 4.1444121
Omega: 22.675908 8.3409576
omega: 296.65769 8.7722517
chi2(PA,sep): 148.25449
chi2red: 1.9253829
chi2(x,y): 148.66500
Number of iterations: 5
```

Plot the visual orbit:

Compile the code:

```
IDL> .r plot_orbit

IDL> plot_orbit,'sample_vb_data',42
Do you want to fit linear motion in RA/DEC (y/n)?
: n
If you would like to enter P,T,e and then have the program do a
least squares fit for a,i,Omega,omega, enter "pte" at prompt.
If you would like to enter all the orbital elements,
(P,T,e,a,i,Omega,omega) then enter "all" at the prompt
: all
Enter P T e a i W w (period, time of periastron passage, eccentricity):
: 41.385217 1981.3805 0.34021606 93.334535 141.85568 22.675908 296.65769
Do you want to plot computed position of binary (y/n)?
: n
Do you want to connect computed and measured positions with a line (y/n)?
: y
Do you want to plot an additional point (y/n)?
: n
Do you want to plot another orbit (y/n)?
: n
Enter plot title:
:
chi2 148.66500
```

The orbit plot is as saved as temp.eps and residuals are saved as temp_res.eps.

Example 3: Spectroscopic Binary Orbit

Fit double-lined spectroscopic orbit (using sigma ori as an example):

```
IDL> fit_velocity_sb2,'sb2_sigori_simondiaz15_hjd',90
Enter P,T,e,K1,K2,omega,Vsys:
: 143.198 56597.623 0.7782 71.9 95.2 199.98 31.10
Vary each orbital element?
For each element, enter 0 to hold element fixed, 1 to vary:
[P,T,e,K1,K2,omega,Vsys]
: 1 1 1 1 1 1 1
Final values:
Orbital Element Error
P: 143.19830 0.0034275749
T: 56597.651 0.018387754
e: 0.77654302 0.00084953986
K1: 72.338984 0.24155389
K2: 94.926165 0.21253425
omega: 200.28115 0.25779794
Vsys: 31.484528 0.19765699
Final chi2: 172.20817
Reduced chi2: 0.99542294
Number of iterations: 3
```

Plot SB2 RV orbit:

```
IDL> plot_velocity_sb2,'sb2_sigori_simondiaz15_hjd',90
Enter P,T,e,K1,K2,omega,Vsys:
: 143.19830 56597.651 0.77654302 72.338984 94.926165 200.28115 31.484528
```

The RV plot and residuals are saved as temp_res2.eps

Example 4: Simultaneous Visual + Spectroscopic Orbit

In this example the visual binary uncertainties are given as error ellipses. If you have uncertainties only in PA and sep (rather than full error ellipses) then use `fit_orbit_vbsb2.pro` instead of `fit_orbit_vbsb2_ell.pro`:

```

IDL> fit_orbit_vbsb2_ell,'vb_sigori_mirc_covar_avg_scale',9,'sb2_sigori_simondiaz15_hjd',90
Enter P,T,e,a(mas),i,Omega,omega,K1(km/s),K2(km/s),Vsys:
: 143.19830 56597.651 0.77654302 4.3035061 56.890301 6.2383883 200.28115 72.338984 94.92616
Vary each orbital element?
For each element, enter 0 to hold element fixed, 1 to vary:
[P,T,e,a,i,Omega,omega,K1,K2,Vsys]
: 1 1 1 1 1 1 1 1 1 1
Final values:
Orbital Element Error
P: 143.19930 0.0034234276
T: 56597.643 0.015481179
e: 0.77787551 0.00058277759
a(mas): 4.2832378 0.0042855226
i: 56.504395 0.12131589
Omega: 6.7319113 0.10944250
omega_A: 200.00522 0.16236887
K1(km/s): 72.373668 0.24085550
K2(km/s): 95.044796 0.20956579
Vsys: 31.484641 0.20014098
Final chi2: 196.36829
Reduced chi2: 1.0445122
Number of iterations: 3

```

As in the previous examples, orbit plots can be made using `plot_orbit.pro`, `plot_orbit_ell.pro`, `plot_velocity_sb1.pro`, and `plot_velocity_sb2.pro`. Each of the plotting routines should be compiled using ".r program_name" before running.

Acknowledgments

For the two chi2 grid search procedures (`test_chi.pro` and `hartkopf_mc.pro`) please reference Schaefer et al. (2006, AJ, 132, 2618) (<https://ui.adsabs.harvard.edu/abs/2006AJ....132.2618S/abstract>).

For the Newton Raphson procedures for visual orbits (`newt_raph.pro` and `newt_raph_ell.pro`) and simultaneously fitting visual and spectroscopic orbits, please reference Schaefer et al. (2016, AJ, 152, 213) (<https://ui.adsabs.harvard.edu/abs/2016AJ....152..213S/abstract>).

Bootstrap (<http://twitter.github.io/bootstrap/>) is a front-end framework of Twitter, Inc. Code licensed under MIT License. (<https://github.com/twbs/bootstrap/blob/master/LICENSE>)

Font Awesome (<http://fontawesome.github.io/Font-Awesome/>) font licensed under SIL OFL 1.1 (<http://scripts.sil.org/OFL>).