# Exoplanet Yield Calculator: A Primer

Christopher C. Stark[1], Aki Roberge[2]

## 1. Introduction

At the 2014 Sagan Summer Workshop you will have the opportunity to experiment with calculating the exoplanet yield for a direct imaging mission. You will define the type of exoplanets you are interested in imaging as well as the telescope and instrument parameters. From these basic parameters, you will be able to calculate an expected yield of V band detections of exoplanets as well as investigate how the yield responds to changes in mission parameters. You will also be provided with an EXCEL mission cost estimator so that you can investigate the economical trade-offs for different mission parameters.

A direct imaging mission requires blocking out the bright stellar light with some form of high-contrast imaging technology such that faint nearby exoplanets can be detected. All high-contrast imaging instruments necessarily block out a portion of the sky centered on the star, within which planets cannot be detected. The characteristic angular scale of this region is referred to as the inner working angle (IWA) of the instrument.

When observing a given star, there is a non-zero probability that the planet you wish to observe is within the IWA and unobservable. The probability that the exoplanet is outside of the IWA is referred to as obscurational completeness. In addition, the planet could be in crescent phase and too dim to observe, or have a lower albedo than expected. The probability that the planet is bright enough to observe is referred to as photometric completeness. The code you will be running uses these probabilities to calculate the chance of observing a given exoplanet around a given star if that exoplanet exists.

These codes are often referred to as completeness calculators. Because of their simplicity and numerical speed, this type of yield calculator is a critical component to the design reference mission (DRM) simulator for future exoplanet imaging missions. You will be using a recent version of a DRM code that was developed at NASA Goddard Space Flight Center to inform the design of the Advanced Technology Large-Aperture Space Telescope (ATLAST).

---

[1]NASA Goddard Space Flight Center, Exoplanets & Stellar Astrophysics Laboratory, Code 667, Greenbelt, MD 20771; christopher.c.stark@nasa.gov

[2]NASA Goddard Space Flight Center, Exoplanets & Stellar Astrophysics Laboratory, Code 667, Greenbelt, MD 20771; aki.roberge@nasa.gov

This code has been adapted for the goals of the Sagan Summer workshop, and some of its features have been simplified to enhance the learning experience. This primer is intended to provide a brief overview of how the code works and how to run the code.

## 2. How the DRM Code Works

There are four fundamental steps to calculate an exoplanet yield using a completeness calculator: target list selection, completeness calculation for each star, exposure time calculation for each star, and yield maximization.

### 2.1. Target list

You will be using stars within 50 pc from the *Hipparcos* catalog. Though this input catalog has not yet been vetted in detail, we have already cross-referenced the target list with the *Hipparcos* Double & Multiples Catalog and the Washington Double Star Catalog to remove stars with companions within $10''$, leaving a total of 5257 potential targets. For each target star, we calculated stellar luminosity from the *Hipparcos* parallax data, *Hipparcos* Johnson $V$ and $B-V$ values, and bolometric corrections given by Equations 1 and 2 from Turnbull et al. (2012). All necessary stellar parameters for the target list will be provided.

As discussed in Turnbull et al. (2012), the *Hipparcos* catalog is incomplete for late type stars. Since late type stars likely have more compact planetary systems, the *Hipparcos* incompleteness may artificially underestimate exoplanet yields for large apertures with small inner working angles.

### 2.2. Completeness calculator

For exoplanet yield, "completeness" is defined as the chance of observing an exoplanet around a given star if that exoplanet exists (Brown 2004). The chance of observing an exoplanet is therefore the product of completeness and $\eta_{\rm planet}$, the fraction of stars that host such a planet. For an exoplanet to be observable, it must appear exterior to the IWA of the instrument, interior to the outer working angle (OWA), and must be bright enough to observe at a given SNR within a given exposure time. Because we will know little about the majority of planetary systems prior to imaging them, completeness is calculated via a Monte Carlo simulation.

The code provided calculates the completeness for each star in the target list given a set of planet and mission parameters. It distributes a large number of synthetic exoplanets around each star in the target list, sampling over all possible orbits, phases, and orientations, and calculates the projected angular separation as well as the planet-to-star flux ratio. For each star, the code then calculates the fraction that are exterior to the IWA, interior to the OWA, and bright enough to observe at the designated SNR within a given exposure time.

You will define the parameters of the exoplanet you are interested in imaging—specifically, the planet radius, geometric albedo, range of semi-major axes, and range of orbital eccentricities. For simplicity, we have predefined 4 types of planets for you to investigate: "Earths," "Jupiters," "Neptunes", and "Warm mini Neptunes." Because this code is primarily intended to calculate exoEarth yield, the code scales the semi-major axes of the planets for each star such that the incident bolometric flux is constant (i.e. the code roughly keeps exoEarths in the habitable zones of every star).

### 2.3. Exposure time calculator

Assuming negligible read noise and dark counts, the exposure time required to image a planet is given by

$$\tau = \mathrm{SNR}^2 \left( \frac{\mathrm{CR_p} + 2\,\mathrm{CR_b}}{\mathrm{CR_p}^2} \right), \tag{1}$$

where SNR is the signal to noise ratio desired for the planet, $\mathrm{CR_p}$ is the photon count rate for the planet, and $\mathrm{CR_b}$ is the photon count rate for the background (Brown 2005). The factor of two in front of the background count rate reflects the assumption of a background subtraction.

The code calculates an exposure time for *every synthetic exoplanet around every star*. The planet's count rate is determined from the planet-to-star flux ratio calculated by the completeness calculator and the apparent V band magnitude for each star in the target list, as well as the telescope and instrument parameters provided.

The code assumes three dominant sources of background flux: leaked stellar light not canceled out by the high-contrast instrument, local zodiacal light, and exozodiacal light. The amount of leaked starlight is controlled by the instrument contrast, $\zeta$, as well as the apparent V band magnitude of each star in the target list. The local zodiacal light is calculated for each star based upon the ecliptic latitude. The exozodiacal count rate is calculated assuming a constant optical depth of dust around each star in units of "zodis," where 1 "zodi" is the V band optical depth of dust in the solar system at 1 AU, as viewed from afar.

### 2.4. Yield Maximization

The estimated exoplanet yield for a mission is equal to $\eta_{\mathrm{planet}}$ times the total completeness obtained for all observed stars within the assumed mission lifetime. Therefore, to calculate exoplanet yield, one must decide which stars to observe, how long to observe them, and in what order, necessitating a target prioritization metric. The code you will be using calculates the benefit-to-cost ratio of each star. The benefit is calculated as completeness $C$, the chance of observing an exoplanet around a given star if that exoplanet exists, and the "cost" is the exposure time $\tau$.

For each star, the code sorts the synthetic exoplanets by exposure time to obtain the completeness and benefit-to-cost ratio as a function of exposure time. The top panel in Figure 1 shows an example exoEarth completeness curve for HIP 54035 using an 8 m aperture mission with a coronagraph. This curve tells us the ultimate single visit completeness for the star in the limit of $\tau \to \infty$ is $\approx 0.87$; 13% of potential exoEarths are obscured and unobservable.

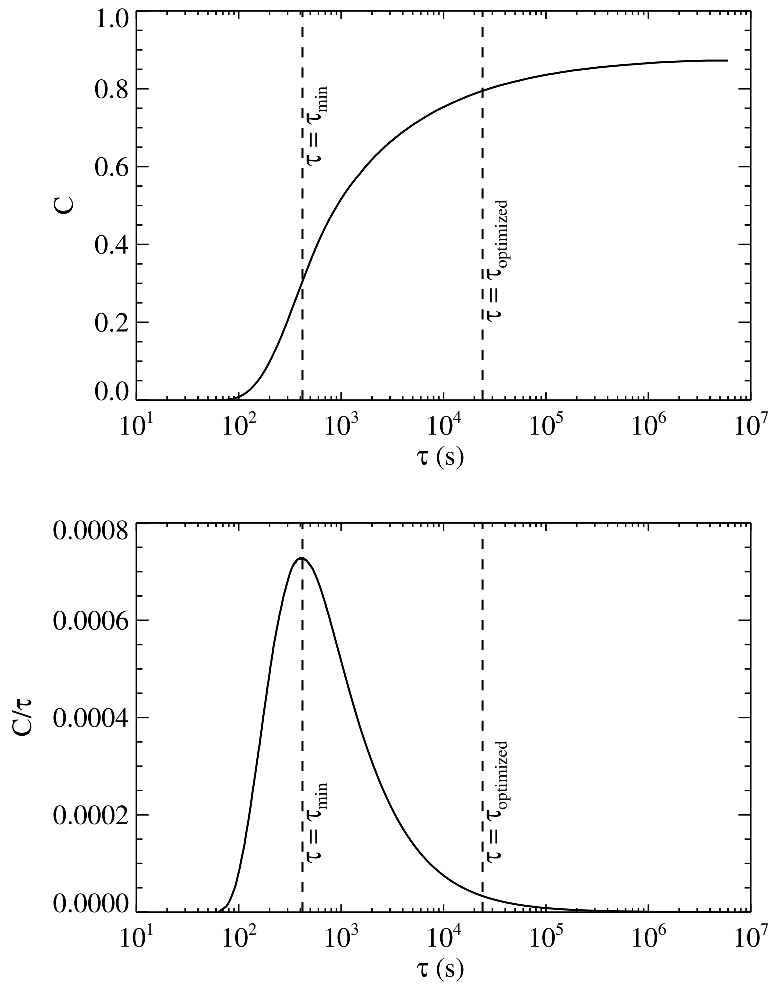


Fig. 1.— *Top*: Single-visit completeness curve for HIP 54035, calculated for our baseline mission. *Bottom*: Prioritization metric (benefit-to-cost ratio) curve. The most efficient observation time occurs at $\tau = \tau_{\rm min}$. The exposure time that helps maximize mission yield occurs at $\tau = \tau_{\rm optimized}$.

The bottom panel in Figure 1 shows the benefit-to-cost ratio as a function of exposure time for this star. The peak of this curve represents the most efficient exposure time for this star, i.e. the time at which we get the most "bang for our buck." Observing for a shorter time would be wasteful since the efficiency and completeness are both low. Therefore, the code sets the exposure time for each target, $\tau_i = \tau_{{\rm min},i}$. The code then prioritizes targets by $C_i(\tau_{{\rm min},i})/\tau_{{\rm min},i}$ and selects the highest priority targets whose total exposure time fits within the budget of the mission.

Although the code has maximized the observational efficiency of each target and prioritized the target list, it has not necessarily maximized the mission yield. The lowest priority target selected has a long exposure time and relatively low completeness—this time may be better spent on another target. The last step of code examines the prioritized target list in reverse order and gives the exposure time from the lowest priority targets to targets that would better increase the mission yield. The result is a prioritized target list with exposure times assigned that maximize the exoplanet yield using the astrophysical and mission parameters assumed. *The code effectively "tunes" the mission to maximize exoplanet yield for the type of planet you're interested in imaging.*

## 3. Limitations of the Code

You should be aware of the fact that this simplified version of the DRM code has several significant limitations. First, it is only valid at V band ($\lambda = 0.55$ $\mu$m). While the code can approximately estimate the impact of taking high resolution spectra on the yield, you cannot adjust the primary wavelength of detection.

Second, this version of the code models single visits to each star and does not handle revisits. For planets with orbital periods $\lesssim$ the mission timescale, revisiting a target at a later time, after the planets have moved along their orbits, can enable the detection of planets that were initially undetectable. Thus the single-visit yield calculated by this code will underestimate the yield of such planets, possibly by a factor of $\sim 2$.

Finally, this version of the code does not include overheads associated with wavefront control, thermal equilibration of the mirror, etc. Roughly speaking, the overhead time for a coronagraph could be as much as the exposure time for a given star, effectively doubling the required exoplanet science time budget. These sorts of overheads, which should count directly against the exoplanet science time budget, are currently poorly understood and are thus left to their own unconstrained budget.

## 4. How to Run the DRM Code

### 4.1. A basic call to the DRM code

The DRM code you will be using is written in IDL. We will provide you with access to IDL and the DRM code while you are at the workshop. IDL is somewhat similar to MATLAB, and if you have modest coding experience, you should be able to pick up the basics very quickly. We will also provide you with sample scripts to run.

After starting IDL, you'll need to load a file that contains all of the necessary procedures. To do so, type

```
IDL> restore,filename='exoplanet_yield.sav',/verbose
```

You should see a list of files and procedures that were loaded. To run the yield calculator, type

```
IDL> yield_calculator, targets, nstars, nplanets, ttotal, planet='earth',
verbose=1
```

In the example above, you are sending `yield_calculator` 4 variables that will contain output from `yield_calculator`. Order of the variables matters, but names do not and they do not need to be defined beforehand.

`targets` will contain the output prioritized target list. `targets` is an array of length `nstars`, with each entry a structure data type. `targets[0].lstar` references the luminosity of the highest priority star selected for observation, while `targets[nstars-1].lstar` references the luminosity of the lowest priority star selected for observation. Other tags of interest include HIP identifier (`.HIP`), distance in pc (`.dist`), apparent and absolute V magnitude (`.V_mag`, `.M_V`), stellar type (`.type`), optimized exposure time (`.texpose`), and the estimated number of planets detected around this star (`.nplanets`).

`nstars` is a scalar equal to the number of observed stars, `nplanets` is a scalar equal to the total number of exoplanets detected, and `ttotal` is the total exposure time used for detection.

`yield_calculator` requires one mandatory keyword, `planet`. You must define the type of planet for which you'd like to optimize the mission, provided as a string. Possible values are `'earth'`, `'jupiter'`, `'neptune'`, and `'warmminineptune'`.

## 4.2. A customized call to the DRM code

Using the example call to `yield_calculator` shown in the previous section, the code will revert to its default values for all astrophysical and mission parameters. You will want to change these parameters to define your mission. You can change any of the assumed parameters by using keywords. Keywords are placed after the output variables in the example above, like in the following example

```
IDL> yield_calculator, targs, ns, np, ttot, planet='earth', keyword1=value1,
```

`keyword2=value2`

Order of keywords does not matter. The available keywords are listed in Table 1.

### 4.3. Two modes of imaging & realistic constraints on instrument parameters

The code handles two types of high-contrast imaging technologies: an internal coronagraph and an external occulter, a.k.a. starshade. These two technologies have very different properties. Internal coronagraphs block starlight after it has been collected by the telescope, while starshades fly millions of meters away and block the starlight before it reaches the telescope. Internal coronagraphs have modest overhead times during which starlight is supressed and no other science can be performed. Starshades have relatively short overheads, but long slew times between stars during which non-exoplanet science can be performed. Further, coronagraphs have poor facility throughput ($\sim 0.2$), while starshades could have facility throughputs ($\gtrsim 0.4$).

By default, the code operates in the internal coronagraph mode. To define your coronagraph instrument, simply set the `contrast`, `iwa`, and `owa` keywords. Here we provide reasonable constraints on these parameters.

Near-term coronagraphs come close to achieving contrasts of $10^{-9}$ (at 20% bandwidth, as assumed by this code). Future coronagraphs may achieve contrasts of $10^{-10}$ or at best, $10^{-11}$. The OWA of a coronagraph is determined by the size of the deformable mirror used—roughly speaking, values of 15–20 $\lambda/D$ are reasonable (the code defaults to $15\lambda/D$). The typical IWA of current coronagraphs is $\sim 3\lambda/D$. Future coronagraphs may achieve $\sim 2\lambda/D$.

To switch to starshade mode, simply set the `starshade` keyword equal to 1. For this workshop, we have kept the functionality of starshade mode the same as coronagraph mode, i.e. you must simply set the `contrast` and `iwa` keywords. The OWA of a starshade is essentially unlimited, so the code automatically sets the OWA to a large value.

The contrast and IWA of a starshade are much different than that of a coronagraph. The contrast and IWA are controlled by the size of the starshade and it's distance from the telescope. For simplicity, the code automatically calculates the approximate starshade diameter and distance based upon the input contrast and IWA desired. The contrast controls the Fresnel number of the starshade, which is converted into a distance. Higher contrast moves the starshade farther away. The diameter of the starshade is then calculated by multiplying the distance by the IWA.

For this workshop, the code approximates the slew time of a starshade as proportional to the distance between the telescope and starshade, with a nominal value of 1 week at a distance of 37 Mm. Slew time does not count against the exoplanet science budget, since other science can be performed during this time, but the total exoplanet science time plus slew time must fit within the total mission lifetime. Starshades also have limited fuel. For this workshop, all starshades are limited to 100 observations. As you will find, the yield of a starshade mission is typically limited by these two factors.

This version of the code does not budget for overheads associated with wavefront control, thermal equilibration of the telescope, etc. Thus, the total exoplanet science time is effectively a total exoplanet integration time.

A less intuitive instrument parameter is the contrast noise floor, i.e. there exists a contrast limit (or dimmest planet) beyond which you cannot detect planets regardless of integration time. It is generally thought that via post-processing methods, one can detect planets 10 to 100× fainter than the raw contrast as long as the required photons exist for the desired SNR. To adjust this value, set the `limiting_magnitude` keyword. By default, `limiting_magnitude` is set to $1/10^{\mathrm{th}}$ of the raw contrast, such that `limiting_magnitude` $= -2.5 \times \log_{10}(0.1 \times \text{contrast})$.

### 4.4. Examples

Here are a few example calls to the DRM:

Suppose we want the yield of exoEarths for a flagship mission with an 8 m aperture using a future internal coronagraph. Since we're interested in exoEarths, we can simply set planet='earth.' Future coronagraphs may have contrasts $\sim 10^{-10}$, IWAs $\sim 2\lambda/D$, and outer working angles $\sim 20\lambda/D$. Such a call to `yield_calculator` might look like

```
IDL> yield_calculator, targs, ns, np, ttot, planet='earth', diam=8, contrast=1e-10,
iwa=0.2, owa=2.0, verbose=1
```

where IWA and OWA must be provided in arcseconds ($\propto \lambda/D$, and $\lambda = 0.55\ \mu$m). The calculation should finish in about a minute. This is a relatively long run time, due to fact that such a large aperture and small IWA can potentially observe a large number of stars.

The code runs much faster if we consider a probe-class mission with a 1.1 m aperture:

```
IDL> yield_calculator, targs, ns, np, ttot, planet='earth', diam=1.1, contrast=1e-10
iwa=0.2, owa=2.0, verbose=1
```

The code should take just a few seconds. You'll notice in this case that the total exoplanet science time is less than the 1 year that was budgeted—an artifact of poor statistics for missions that have very low yields. This can largely be overcome by setting the `nt` keyword to a value smaller than the default value of 1000—e.g., `nt = 100` typically solves this issue. Rerunning the above code with `nt = 100` shows that the yield does not change dramatically, though, and that this is a minor issue.

To simulate a probe-class mission with a starshade, a call might look like

```
IDL> yield_calculator, targs, ns, np, ttot, planet='earth', diam=1.1, starshade=1,
contrast=1e-11, iwa=0.1, verbose=1
```

Notice that in this case the code outputs additional information about the starshade.

If we want to take an $R = 70$ spectra on every detected exoEarth, the call would look like

```
IDL> yield_calculator, targs, ns, np, ttot, planet='earth', diam=1.1, starshade=1,
contrast=1e-11, iwa=0.1, specresolvingpower=70, verbose=1
```

Here a lower number of exoEarths are detected because a portion of the exoplanet science budget is devoted to characterization.

### 4.5. Calling DRM from a wrapper

If you'd like to obtain results for some known set of parameters, you could automate your calls to the DRM code via a wrapper. Be aware that you should estimate your run times prior to running the wrapper for a large number of scenarios.

You should have a sample wrapper called `sample_wrapper.pro`. Within it, you will see how to loop over several parameters such that you can call `yield_calculator` multiple times for vectors of values. Feel free to use, ignore, or edit your copy of the wrapper. To compile it, type

```
IDL> .r sample_wrapper.pro
```

To run it, type

```
IDL> sample_wrapper
```

If you edit the `sample_wrapper` file, you must recompile it prior to running it again.

### 4.6. Plotting results

Hopefully you will want to plot your results. What should you plot? Anything you want to communicate that you have learned. The most obvious plot is yield as a function of instrument/telescope parameters. Other interesting plots could include the average or maximum exposure time per target as a function of instrument/telescope parameters to see whether the required exposure times are realistic (remember, these values are returned in the `targets.texpose` vector), or the stellar luminosity or distance as a function of priority ranking to examine stellar selection effects (remember, `targets[0].dist` is the distance of the highest priority target). If you are also using the EXCEL cost module, you could plot yield per dollar.

You are welcome to make your plots any way you desire. If you would like to make plots in IDL, and are unfamiliar with how to plot in IDL, here are a few tips. A plot command in IDL looks something like the following

```
plot, x, y, xs=1, ys=1, xrange=[xmin,xmax], yrange=[ymin,ymax], xtitle='Hello',
ytitle='World', xlog=1, ylog=1, psym=4
```

where `x` and `y` are vectors of data to be plotted, `xs=1` specifies the x-axis conforms to your specifications, `xrange` sets the upper and lower limits of the x-axis, `xtitle` provides the text along the x-axis, `xlog` sets the x-axis to a logarithmic scale, and `psym` sets the symbol used for plotting. If you would like a line instead of symbols, you should ignore the `psym` keyword and use the `linestyle` keyword instead.

You should also have available a sample plot maker, called `sample_plot.pro`. This file creates a fancy color encapsulated postscript plot to show you what is possible. Feel free to use, ignore, or edit this file.

### 4.7. Warnings, Troubleshooting, & "Help, Please!"

If the parameters defining your mission are relatively poor, i.e. small telescope aperture with low contrast and large IWA, you may run into very low yield numbers. In this situation, the number of Monte Carlo points simulated internally by the code may need to be increased to provide accurate results, or the number of time bins used to redistribute exposure time may need to be decreased. To decrease the number of time bins, use the `nt` keyword (default value is $10^3$). To increase the number of simulated planets, which will increase the run time of the code, use the `np` keyword (default value is $10^5$).

If you don't understand something, or don't know how to do something, feel free to ask.

You have very limited time for this exercise, so no question is too stupid!

## REFERENCES

Brown, R. A. 2004, ApJ, 607, 1003

—. 2005, ApJ, 624, 1010

Turnbull, M. C., Glassman, T., Roberge, A., et al. 2012, PASP, 124, 418

This preprint was prepared with the AAS LaTeX macros v5.2.

Table 1. Keyword list

| Keyword | Units | Default value | Description |
|---|---|---|---|
| `planet` | | N/A | ‘earth’, ‘jupiter’, ‘neptune’, or ‘warmminineptune’ |
| `verbose` | | 0 | Set this flag to 1 to print results |
| `diam` | m | 8 | telescope diameter |
| `contrast` | | $10^{-10}$ | contrast achieved by high-contrast instrument |
| `iwa` | arcsec | $2\lambda/D$ | inner working angle |
| `owa` | arcsec | $15\lambda/D^{*}$ | outer working angle |
| `starshade` | | 0 | Set this flag to 1 to switch from coronagraph to starshade mode |
| `nexozodis` | “zodis” | 3 | Amount of exozodiacal dust around all stars |
| `mission_lifetime` | years | 5 | Total time of mission |
| `exoplanet_science_fraction` | | 0.2 | Fraction of mission devoted to exoplanet science |
| `eta_planet` | | 0.1 | Fraction of stars hosting that type of planet |
| `SNR` | | 10 | signal-to-noise ratio required for exoplanet detection |
| `limiting_magnitude` | mags | $-2.5 \times \log_{10}(0.1 \times \text{contrast})$ | dimmest planet detectable at SNR |
| `specresolvingpower` | | 5 | spectral resolving power ($\lambda/\Delta\lambda$) |

*In starshade mode, OWA is effectively infinite.