Fast Linearized Coronagraph Optimizer (FALCO) I: A software toolbox for rapid coronagraphic design and wavefront correction

A J Eldorado Riggs[a], Garreth Ruane[b], Erkin Sidick[a], Carl Coker[a], Brian D. Kern[a], Stuart B. Shaklan[a]

[a]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109

[b]California Institute of Technology, Pasadena, CA 91125

ABSTRACT

The Fast Linearized Coronagraph Optimizer (FALCO) is an open-source toolbox of routines for coronagraphic focal plane wavefront correction. The goal of FALCO is to provide a free, modular framework for the simulation or testbed operation of several common types of coronagraphs. FALCO includes routines for pair-wise probing estimation of the complex electric field and Electric Field Conjugation (EFC) control, and we ask the community to contribute other wavefront correction algorithms. FALCO utilizes and builds upon PROPER, an established optical propagation library. The key innovation in FALCO is the rapid computation of the linearized response matrix for each deformable mirror (DM), which facilitates re-linearization after each control step for faster DM-integrated coronagraph design and wavefront correction experiments. FALCO is freely available as source code in MATLAB at *github.com/ajeldorado/falco-matlab* and will be available later this year in Python 3 at *github.com/ajeldorado/falco-python*.

Keywords: wavefront control, wavefront estimation, coronagraph, high-contrast imaging, PROPER, modeling

1. INTRODUCTION

Over the past two decades, there has been much progress in closing the technology gaps needed to image exoplanets in reflected light with future ground- and space-based telescopes. Coronagraph design, fabrication, and modeling have advanced greatly as part of the development for obstructed and/or segmented apertures such as the Wide-Field Infrared Survey Telescope (WFIRST),[1] the Large Ultraviolet Infrared telescope (LUVOIR),[2] and the three upcoming 30-meter-class ground telescopes. At the same time, a number of testbeds have been built across the United States and Europe to test these coronagraphs and develop new algorithms for wavefront sensing and control.

Fourier optics is necessary to model the starlight suppression regions, or "dark holes", of a coronagraphic image correctly at high contrast ratios of a million or more. The astronomical community commonly uses two free, established software libraries for scalar wave optics propagation: PROPER[3] and Physical Optics Propagation in Python (POPPY).[4] PROPER, originally released in IDL in 2007 and in Python and MATLAB in 2017, is a collection of routines to set up a high-fidelity optical model and perform the Fresnel propagations within it. POPPY,[4] available since 2012, is as an object-oriented Python library with some overlapping capabilities compared to the newly available translation of PROPER in Python. Por et al.[5] have just released a more mutli-purpose software package called High Contrast Imaging for Python (HCIPy), which has functionality for end-to-end scalar or vector optical propagations, pupil-plane apodizer optimization, wavefront correction, and atmospheric turbulence. A more thorough review of open-source software for coronagraphy is provided by Ruane et al.[6] in these proceedings.

In this paper we present the Fast Linearized Coronagraph Optimizer (FALCO), an open-source software collection to help meet the current needs of the high-contrast imaging community. FALCO's function library

Send correspondence to A.J. Riggs: aj.riggs@jpl.nasa.gov
©2018 California Institute of Technology. Government sponsorship acknowledged.

depends heavily on PROPER and defines many new functions as well. The primary purpose of FALCO is to provide a free, modular framework for the design, simulation, and testbed operation of several common types of coronagraphs. The hope is that this will enable researchers to focus on new technology development instead of having to spend months implementing established coronagraphic wavefront sensing and control (WFSC) techniques for their simulations or testbeds. We therefore provide several fully running examples of coronagraphic WFSC for scientists to learn from and adapt to meet their needs. FALCO is specific to static and quasi-static speckle correction and does not include atmospheric turbulence modeling and correction.

The other main purpose of FALCO is to provide a rapid wavefront control algorithm to advance the state-of-the-art in coronagraphs using deformable mirrors (DMs) as integral components. The two well-known examples of this are the hybrid Lyot coronagraph (HLC)[7,8] and active compensation of aperture discontinuities (ACAD)[9] with optimized stroke minimization (ACAD-OSM).[10,11] Both HLC design and ACAD-OSM rely on wavefront control methods using the linearized response matrix, also known as the control Jacobian, between the electric fields at the DMs and the science detector.[12,13] These control Jacobians must be re-computed several times during design because the cumulative DM actuation degrades the accuracy of the previous linearization. The conventional method of obtaining the response of each actuator, in which the entire beam is propagated in large, zero-padded arrays with fast Fourier Transforms (FFTs) through the optical system, is prohibitively expensive and can take many hours or even days to calculate on a single computer for a polychromatic case. Using the strategies detailed in Section 4.1, FALCO reduces the control Jacobian calculation time by orders of magnitude without any loss of accuracy. Sidick et al. (FALCO II)[14] quantify this speed gain and accuracy for a WFIRST-specific case. FALCO's speed has enabled comprehensive DM-integrated design surveys to determine the best configuration parameters for different types of coronagraphs. In these proceedings, the first of these FALCO-enabled studies are presented in papers by Coker et al. (FALCO III)[15] and Ruane et al. (FALCO IV).[16]

In this paper, we provide an overview of FALCO's architecture and key algorithms. The specifics of the code, which will evolve over time, are left to the user manual in order to reflect the most recent software version. Section 2 of this paper describes the relationship between FALCO and PROPER and the development status of FALCO. Next, Section 3 describes the MATLAB version's software framework and the three types of optical models used. Section 4 details the various methods that provide rapid optical propagation in FALCO. Finally, Section 5 presents results from several different trials of FALCO.

1.1 Availability

FALCO is freely available online as open-source code in MATLAB at *github.com/ajeldorado/falco-matlab*. A Python 3 version is in development and will be available later this year at *github.com/ajeldorado/falco-python*.

2. FALCO CAPABILITIES

2.1 Relation to PROPER

FALCO builds upon and utilizes the optical propagation library PROPER.[3] Table 1 compares their capabilities and development levels. Whereas PROPER is a library of building blocks for constructing and propagating through an optical model, FALCO is a collection of code that provides working examples of WFSC for several coronagraph types. Several key parts of FALCO directly use functions from PROPER, for example the deformable mirror model, the generation of many masks, and the end-to-end, aberrated Fresnel model of a testbed or instrument. By relying on the more mature PROPER library for these critical parts of the WFSC simulations, we greatly decrease the risk of software bugs and can dedicate more time to new features in FALCO.

2.2 Noteworthy Features and Development Progress

FALCO provides or will provide several major new capabilities. The first is working examples of WFSC code for a coronagraphic optical system. The second, but most distinguishing, feature is the rapid calculation of control Jacobian matrices, which are the linearized responses of the deformable mirror(s) in the final focal plane. Compared to conventional methods, FALCO computes the control Jacobians in orders of magnitude less time–but with the same accuracy–using the techniques described later in Section 4.1. Sidick et al.[14] quantify the

Feature	PROPER	FALCO	Comments
Wavefront estimation and control functions.			No ground-based, adaptive-optics sensing and control is included in FALCO.
Deformable mirror (DM) model.			FALCO uses PROPER's DM model.
Operate a testbed at a high level			User must provide low-level, HW-to-SW code
Functions to construct a realistic **testbed model**			FALCO requires the user to provide a "full model" most easily made with PROPER
Running examples of coronagraphs			PROPER provides one demo for a band-limited Lyot coronagraph
Functions to **generate telescope pupils**			FALCO constructs specific pupils with functions from PROPER.
Numerical safety checks in optical propagation functions.			PROPER checks every propagation while FALCO checks for angular spectrum.
User friendliness			FALCO adds more features to PROPER at the expense of user friendliness.
Available in multiple **languages**			FALCO in Python 3.6 is in development.
Functional testing done by developer.			

Table 1. The capabilities and complementarity of PROPER and FALCO.

speed gain and accuracy in the FALCO II paper in these proceedings. The third key feature of FALCO is its ability to run or simulate WFSC experiments on a testbed.

FALCO has been released in its development stage in order to provide some features to the community immediately. The wavefront control and rapid Jacobian calculation code are available at the time of this writing, but the wavefront estimation and testbed control code will be completed later in 2018 for use on the High Contrast Spectroscopy Testbed for Segmented Telescopes (HCST) at the California Institute of Technology.[17] In addition to the single controller available now, electric field conjugation (EFC),[12] we will add a speckle nulling controller.[18] Currently the only available deformable mirror model is for an electrostrictive Xinetics[19] DM. We will soon be adding a model of Boston Micromachines MEMS DMs[20] to PROPER (and thus to FALCO) as well.

3. SOFTWARE ARCHITECTURE

3.1 Organization

The MATLAB implementation of FALCO uses the framework diagrammed in Fig. 1. The data flow and algorithms have been modularized to enable easy switching between testbed and simulation, different coronagraph types, and different estimators or controllers. A main script defines any different parameter values before calling a branching tree of functions to perform WFSC. Any required variables left undefined in the main script are given a default value to ensure that the trial runs, but the user must determine the best default values for their application.

Minimal data is saved to avoid filling storage unnecessarily. The parameters saved right after loading any default values are sufficient to re-generate any masks or other arrays if they are needed after a trial is over. Similarly, after a trial is completed, only the DM commands and performance metrics from each iteration are saved because the starting parameters and DM command history are sufficient data to reproduce the electric field and point-spread function (PSF) at any step using the same model functions.

Example hardware-software interface functions for the DM(s) and detector will be provided in FALCO primarily to show what inputs and outputs they will have. We plan to include the functions we will use to run the

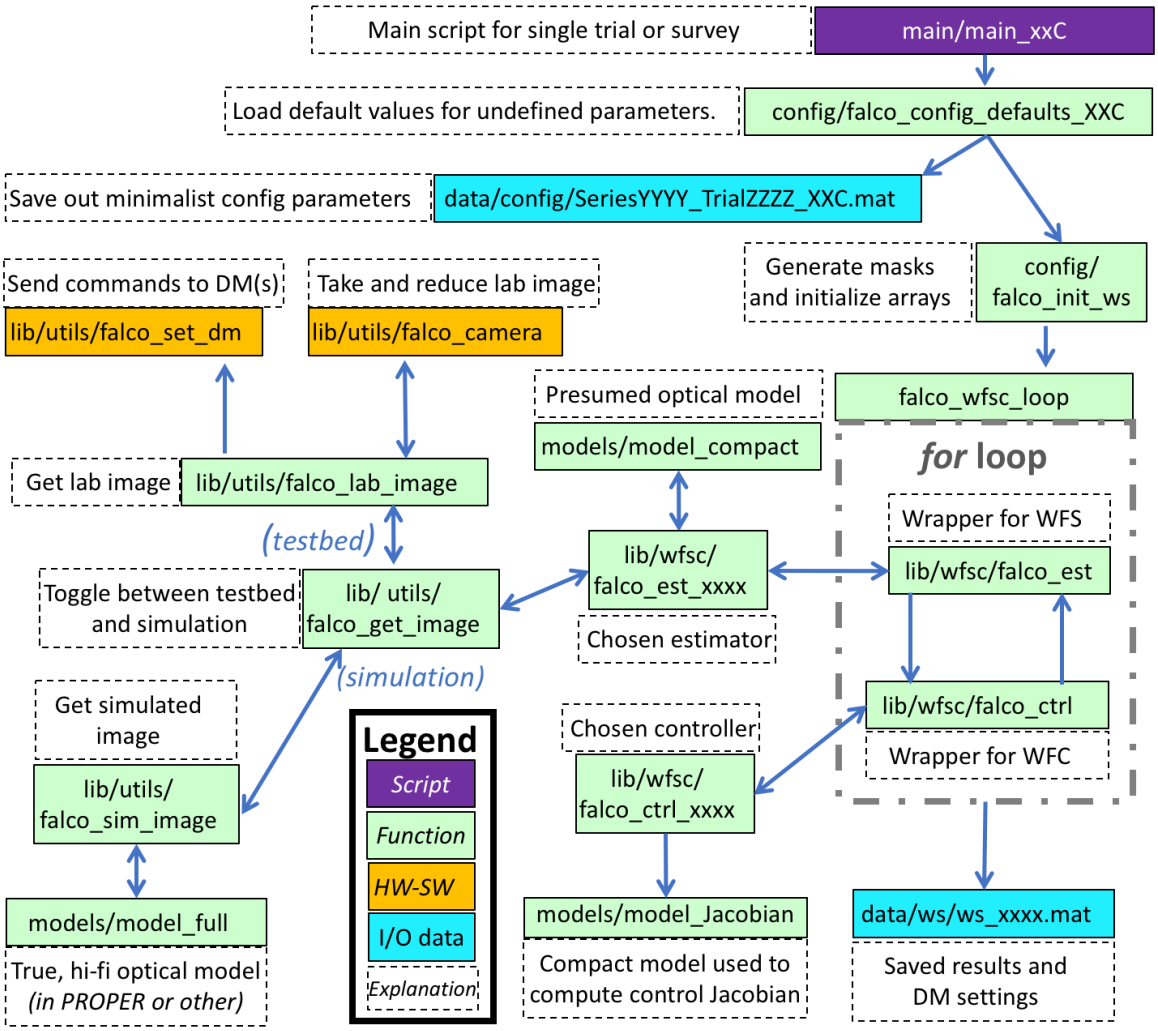


Figure 1. Diagram showing the data flow in FALCO Version 1.0 in MATLAB.

Boston Micromachines DMs in Caltech's High Contrast Spectroscopy Testbed for Segmented Telescopes (HCST), but it remains the user's sole responsibility to adapt those functions to work for their application.

3.2 The Three Types of Optical Models

In FALCO, each coronagraph type has three types of optical models for different situations; we call these the full, compact, and Jacobian models.

3.2.1 Full Model

The full model provides a high-fidelity representation of a coronagraphic testbed or instrument and is therefore used only in simulations rather in actual testbed operations. We recommend using PROPER to construct the full model in order to include realistic optical aberrations and accurate surface-by-surface Fresnel propagations with ease. Except for idealistic design cases, the output electric field from the full model is never used directly. To mimic realistic testbed behavior, the full model is called only by a detector function that returns an intensity image.

It should be noted that at the time of this writing, the example full models included in FALCO Version 1.0 are placeholders useful only for coronagraph design. That is, the current full models have the same simplified

layouts as the compact models lacking aberrations and Fresnel propagations (except between the DMs). The Fresnel-based full model of a testbed or instrument necessarily has more planes and propagations than the compact model. To show how a PROPER full model and a simpler compact model of the same optical system should coordinate, we will provide one or more examples in the next major release of FALCO.

3.2.2 Compact Model

Whereas the full model represents the true physics of a testbed or instrument, the compact model represents the best knowledge of the setup that is available to the estimator and controller. Because it has proven too difficult to measure the aberrations of each optic and incorporate them separately into the compact model at high accuracy, the most common approach is to collapse all the aberrations down into the input pupil plane. The aberration map is found with some form of phase retrieval. Without aberration maps for the powered optics and fold mirrors, there is no loss in accuracy but much time saved by propagating directly between a pupil and focal plane with a Fourier transform. The only Fresnel propagations required in the compact model are between DMs or for any masks non-conjugate to a pupil or focus.

Fig. 2 shows the compact model layout used by default for all the coronagraph types in FALCO. (The default layout represents most current high-contrast testbeds, but the user can re-arrange, remove, or add planes as needed.) There are four pupil and focal planes with shorthand names starting with "P" and "F", respectively. Because there are only re-imaging optics between some pupils, propagations from pupil planes P1 to P2 and P2 to P3 are performed by 180-degree rotations rather than two discrete Fourier transforms (DFTs). However, this requires that the resolution of each pupil plane be the same in the compact model. FALCO allows the two DMs to be anywhere in the collimated space near pupil P2. The E-field is back-propagated to pupil P2 to propagate more easily to plane P3. The methods for propagating efficiently from planes P3 to F3 and F3 to P4 vary depending on the coronagraph type; these are discussed in Section 4.1.

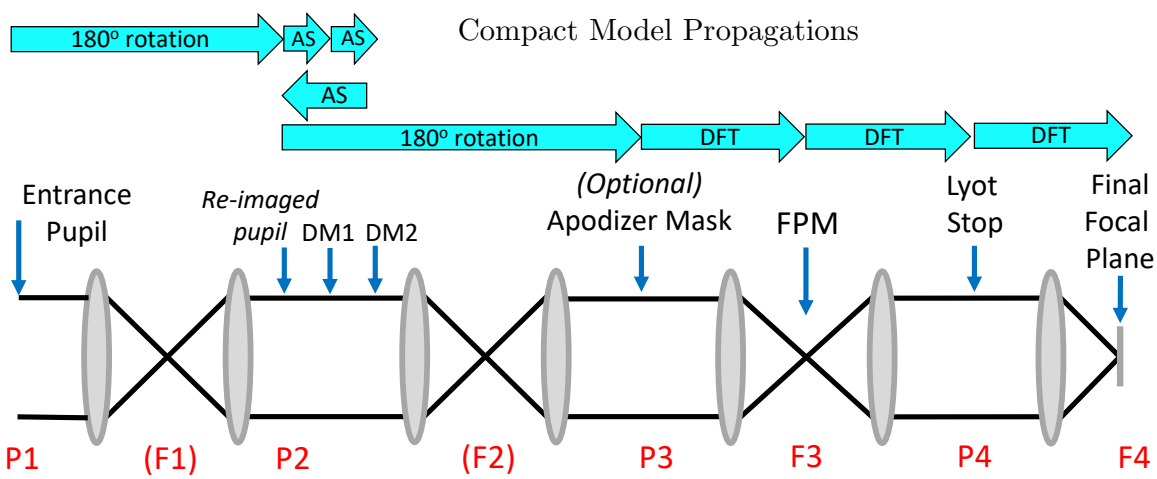


Figure 2. Diagram of the compact model assumed for each coronagraph type. Shorthand names of the planes are written in red below the beam train, where P stands for pupil and F stands for focus. The names of the key optics are pointed out above the beam train in black. The propagations used in the compact model are shown with arrows at the top of the figure, where AS stands for angular spectrum Fresnel propagation and DFT stands for a discrete Fourier transform. Re-imaging a plane is performed simply by rotating the beam by 180 degrees. Measurements of the optical aberrations (such as from phase retrieval) are generally condensed into one plane and are therefore included at the entrance pupil.

3.2.3 Jacobian Model

The third type of model–which gives FALCO its name as a fast linearized optimizer–is the Jacobian model. As the model name suggests, this model is used only for computing the control Jacobian matrices that provide the linearized response of each DM actuator in the final focal plane's E-field. Rather than computing the control Jacobian the conventional, slow way (via differencing E-fields from the compact model with each actuator poked

and unpoked), FALCO computes the Jacobian directly as the first order Taylor series expansion term in the linearization of the DM surface(s). Because anything outside the influence function of the poked actuator is zero, FALCO propagates only the sub-array containing the influence function (with zero padding if necessary for FFTs) as far as possible along the optical path until the full array must be used again. In FALCO, we re-use the same resolutions at each plane in both the compact and Jacobian models so that the intensity normalization constant can be found with the compact model.

4. PROPAGATION METHODS

There are several available methods for performing optical propagations in 2-D Cartesian coordinates. All of them utilize discrete Fourier transforms (DFTs), either as fast Fourier transforms (FFTs) or as matrix Fourier transforms (MFTs).[21,22] In general, FFTs are faster for plane-to-plane, also called angular spectrum (AS), propagation because the input and output planes are at the same size and resolution. AS propagation uses the convolution form of the Fresnel diffraction integral. For direct Fresnel propagation, which uses a single DFT with quadratic phase functions, FFTs are fastest if the full beam is needed at the output plane and resolution is not a concern. If high resolution is needed at both the input and output planes over large areas, then a chirp z-transform (CZT),[23] which uses multiple FFTs, is most efficient. Finally, if only a small area in one or both of the input and output is used, then an MFT is usually most efficient.

In the official PROPER package, the only DFTs are FFTs embedded in the near- and far-field Fresnel propagators. To adjust the focal plane resolution (such as to have the plate scale at a detector the same at all wavelengths), PROPER uses Lanczos interpolation to resize the focal plane. FALCO instead uses an MFT in the final propagation to the focal plane detector in order to avoid potential interpolation errors and reduce runtime; however, a quantitative comparison of speed and accuracy has not yet been performed.

FALCO currently provides five propagation functions. One is for simple re-imaging to the same size and resolution, but rather than performing two DFTs it just rotates the beam 180 degrees while maintaining the correct centering on the array. Two MFT functions are provided, one for going from pupil to focus and another for focus to pupil. A stripped down version of PROPER's `prop_ptp` is used for near-field propagation to and from the DMs. Finally, FALCO contains a high-fidelity propagator for the vortex coronagraph that goes from the pupil plane before the vortex to the pupil after it. This vortex propagator uses two MFTs for going to or from a high-resolution inner focal plane region and a lower resolution full focal plane. The high-resolution inner section is used to sample the vortex's singularity well enough for the numerical noise floor to reach the double-precision limit. The two sections of the vortex are combined using Tukey windows to avoid artifacts from the change in resolution.

4.1 Rapid Calculation of Control Jacobians in FALCO

FALCO uses a number of specialized strategies to reduce the computation time drastically for the control Jacobian. Several of these tactics are common to all coronagraph types and are diagrammed in Fig. 3. The solid-border, blue boxes in perspective view show the location of the square array or sub-array that is propagated away from that plane. The arrows above the beam train show the order and methods of the propagations performed. The entire E-field at the entrance pupil at P1 is re-imaged (via a simple 180-degree rotation) to pupil P2 before being minimally zero-padded and AS propagated to the plane of DM1. (If the Jacobian is being calculated for DM2, then the DM1 surface is applied to the wavefront and the entire, zero-padded E-field is also propagated to DM2.) After the full-array propagation of the whole beam up to a DM, only a sub-array around each DM actuator's influence function is propagated for a few steps. This sub-array propagation (of the first-order Taylor series term linearized about the DM surface) is the single most significant speed-up in FALCO's Jacobian calculation. Note that a backward propagation is performed from DM2 back to pupil P2 so that a simple rotation can be performed from pupil planes P2 to P3. The propagations between pupils P3 and P4 are coronagraph-type dependent and will be explained shortly. Finally, an MFT is used to propagate from a sub-array just encompassing the Lyot stop opening down to the size of the dark hole in the final image plane, F4.

Depending on the radial extent of the focal plane mask (FPM), FALCO uses one of the three methods shown in Fig. 4 to propagate efficiently to and from focal plane F3. The dotted red box at each plane shows the size

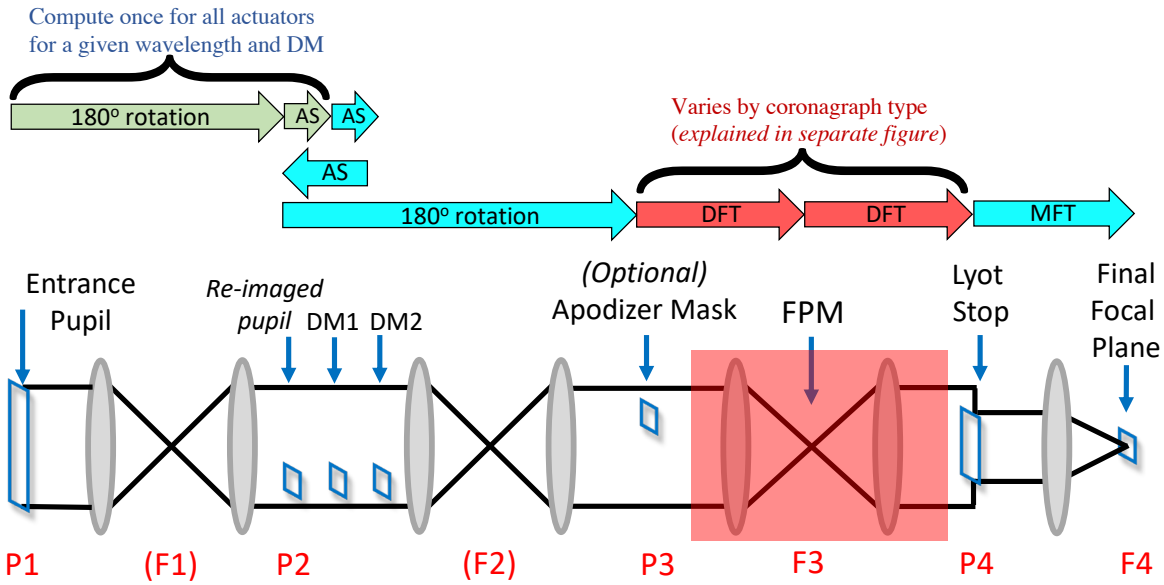


Figure 3. Jacobian model aspects common to all coronagraphs. The only major differences in methodology occur between pupil planes P3 and P4. The blue, projected squares outline the region or sub-region of the beam propagated from that plane. The propagation steps and their types are shown above the beam train.

of the array that is propagated. The dashed cyan circle shows the nominal beam diameter in plane P3, and the filled white circle in plane P4 shows the outer opening of the Lyot stop mask.

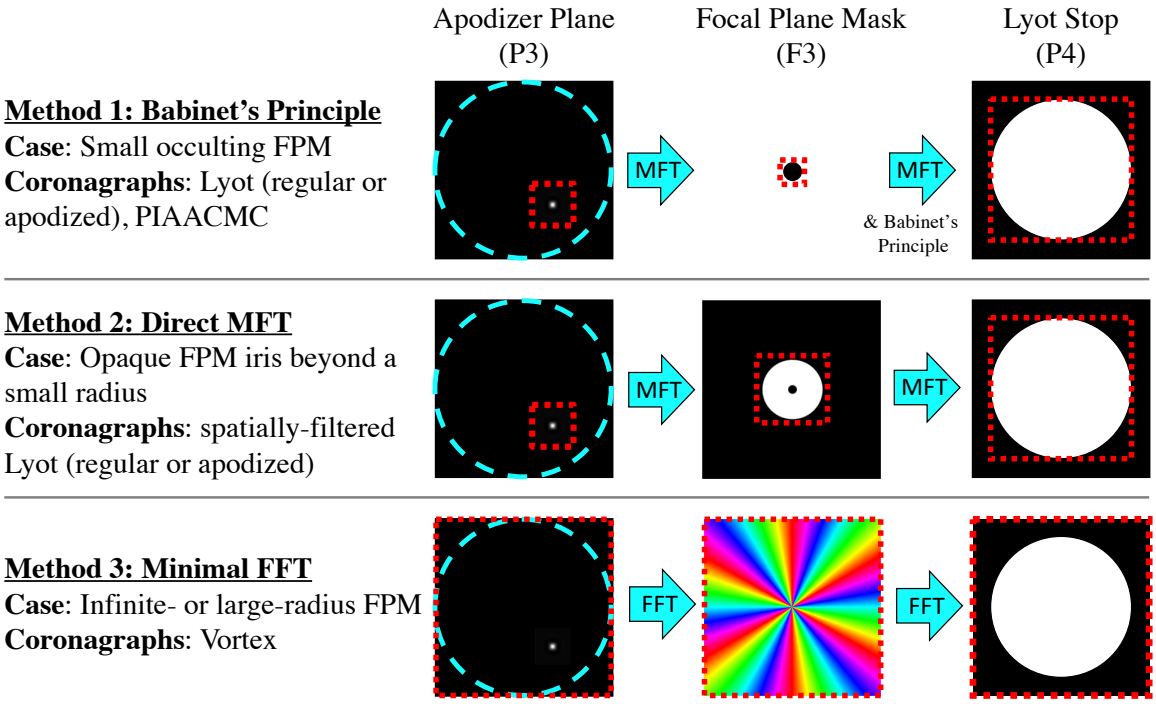


Figure 4. The most efficient Jacobian model propagations to/from the FPM depend on the FPM's spatial extent. MFTs are better for small-extent FPMs such as phase or amplitude spots, and FFTs are better for large- or pseudoinfinite-extent FPMs such as the vortex.

Coronagraphs having a FPM with a small radial extent and uniform transmission (or reflection) outside it can take advantage of the semi-analytical propagation method described by Soummer et al.[21] Some examples are the Lyot coronagraph (FPM = opaque occulting spot), hybrid Lyot coronagraph (FPM = mostly opaque occulting spot with some phase shifting), Roddier-Roddier mask (FPM = π-phase-shifting spot), dual-zone phase mask, and multi-zone phase mask (used in the Phase Induced Amplitude Apodization Complex Mask Coronagraph, or PIAACMC). The semi-analytical trick, which uses Babinet's principle, propagates with MFTs to and from the focal plane F3 only where the small, central part of the FPM is located. This E-field term at pupil P4 is then simply subtracted from a rotated (via re-imaging) version of pupil plane P3's E-field at pupil plane P4.

The second propagation method to/from the FPM in the Jacobian model is for filtered Lyot coronagraphs (FLCs), which spatially filter the focal plane F3 with an outer opaque iris at a relatively small radius (e.g., $\lesssim 30\lambda/D$). FLCs are not common, and the authors' only knowledge of FLC designs are for WFIRST; the apodized FLC has already been named in the literature as a shaped pupil Lyot coronagraph (SPLC).[24] Because the E-field is zero beyond a finite radius at plane F3, the propagation to/from the FPM is most efficient with just MFTs. Whereas Babinet's principle requires plane P4 to have the same resolution as plane P3 for a regular Lyot coronagraph, the direct MFT for an FLC allows the resolution at P4 to be lower. A coarser P4 sampling does not necessarily sacrifice accuracy because the E-field low-pass filtered by the FPM does not need as many points to represent it.

The third and slowest P3-F3-P4 propagation in FALCO's Jacobian model is for (pseudo)infinite-extent FPMs such as the vortex. The E-field must then be computed at all points in the focal plane F3, which spurs us to use FFTs to/from the FPM. The speed-up attained in this case is from zero-padding only the differential E-field around the poked actuator–not of the nominal beam at P3. For simplicity of indexing sub-arrays, we (unevenly) zero-padded the E-field sub-array at P3 to be the original beam size centered on the origin. This easily implemented strategy likely causes excessive zero-padding and therefore wasted time. In the near future, we will further optimize the vortex coronagraph's Jacobian model.

Standard speed-up methods also work for FALCO's Jacobian model. Reducing the resolution at each plane as much as allowable makes a large impact, especially for MFTs. Fine-tuning the resolutions of planes and down-sampling of mask representations is left to the user because it depends on the coronagraph type and performance targets. FALCO can use MATLAB's `parfor` function to parallelize the two most expensive calculations, which are for the control Jacobian calculation and the full model's broadband PSF. We are currently investigating the use of graphics processing units (GPUs) to accelerate the control Jacobian computation in FALCO.

5. EXAMPLES

FALCO Version 1.0 provides the flexibility to perform rapid focal plane wavefront control for various telescopes and coronagraph types. Fig. 5 shows two design examples with (a) a shaped pupil Lyot coronagraph (SPLC) for the original LUVOIR "A" concept[2] and (b) a charge-6 vortex coronagraph (VC) for an off-axis, segmented LUVOIR "B" concept. For these proof-of-concept designs we used 64×64-actuator DMs. The control Jacobian calculations were parallelized into twelve parts, corresponding to the two DMs and six wavelength samples in (a) a 10% bandwidth and (b) a 20% bandwidth, and run on a 12-core workstation. The DM-SPLC optimization completed in five hours after 15 Jacobian calculations, and the DM-VC optimization finished in two hours after five Jacobian calculations. FALCO's relatively short run-times for DM-integrated designs enabled the surveys described in our companion papers Coker et al. (FALCO III)[15] and Ruane et al. (FALCO IV).[16]

Fig. 6 shows a usage of FALCO for wavefront correction with a LUVOIR apodized pupil Lyot coronagraph design provided by Drs. Neil Zimmerman and Roser Juanola-Parramon of NASA Goddard Space Flight Center (GSFC). These calculations were performed to support the study in these proceedings by Juanola-Parramon et al.[25] The pupil masks are shown in Figs. 6(a,c,d), and the random segment phasing errors of the primary mirror are shown in Fig. 6(b). If plotting is enabled, FALCO provides a live update of wavefront control's progress in a figure window. Figs. 6(e) and 6(f) show the progress windows before and after control, respectively.

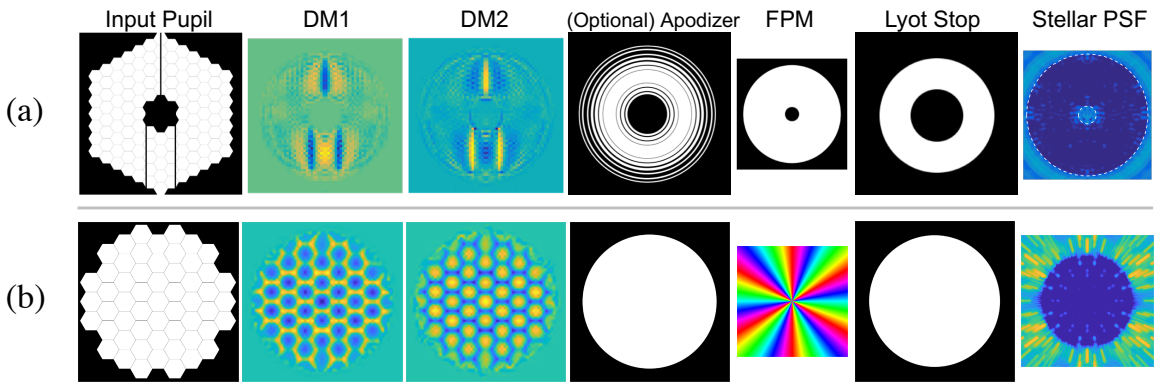
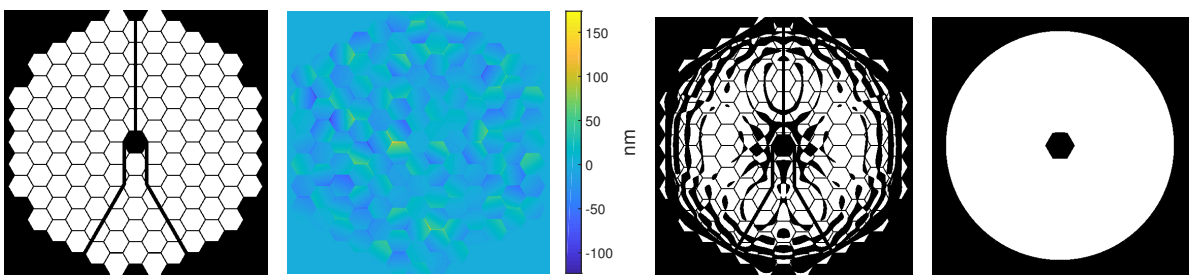
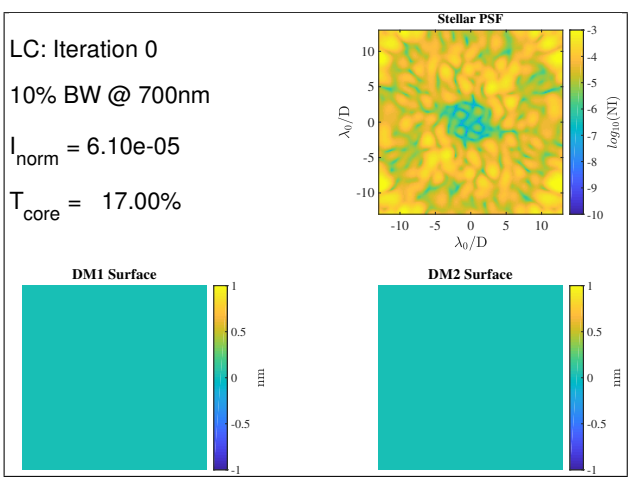


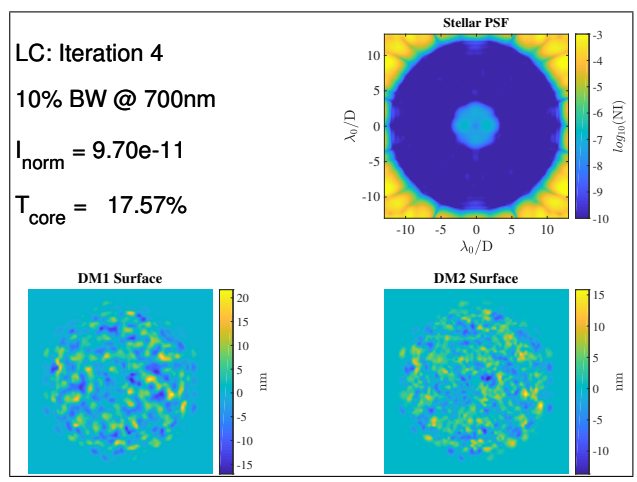
Figure 5. Examples of DM-integrated coronagraph designs computed with FALCO. (a) 10%-bandwidth, DM-augmented SPLC design for the on-axis, LUVOIR "A" concept. (b) 20% bandwidth, DM-augmented vortex coronagraph design for the off-axis LUVOIR "B" concept. The opposite shapes on DMs 1 and 2 indicate that the DMs are apodizing the input pupil's struts and segment gaps. More details are provided by Ruane et al.[16] in these proceedings.



(a) Telescope pupil (b) Segment phasing errors (c) Apodizer (d) Lyot stop



(e) Status window before control



(f) Status window after control

Figure 6. Example with FALCO used to correct (b) static piston-tip-tilt segment alignment errors on (a) a proposed on-axis primary mirror architecture for LUVOIR. The APLC design, including the (c) apodizer and (d) Lyot stop, were provided by Drs. Neil Zimmerman and Roser Juanola-Parramon of NASA GSFC. FALCO reports the progress of wavefront control in a figure window after each correction iteration. The (e) first and (f) last progress windows for this case are shown. BW = spectral bandwidth. I_{norm}=NI= normalized intensity. T_{core} = absolute core throughput.

6. SUMMARY

The Fast Linearized Coronagraph Optimizer (FALCO) is a powerful new software package for high-contrast imaging research. Using and building on the PROPER optical propagation library, FALCO provides wavefront

control routines and examples for several common types of coronagraphs. The defining feature of FALCO is an orders-of-magnitude faster computation–with no loss of accuracy–of the DM response matrix used in focal plane wavefront control algorithms such as electric field conjugation (EFC) and stroke minimization.

We released FALCO this year without all its intended capabilities to undertake pressing design studies for the LUVOIR mission concept. FALCO Version 1.0 is predominantly useful for rapid optimization of DM-integrated coronagraphs. Our series of papers in these proceedings detail the first FALCO-enabled design surveys. Over the next several months, we will add the two major, missing features of FALCO: focal plane wavefront estimation and testbed compatibility. Both features will be released later this year after being implemented on the HCST bench at Caltech. Once updated to include those features, the next version of FALCO will be a powerful tool and flexible software platform, providing the community with rapid wavefront correction code for modeling or running a wide range of coronagraph benches or instruments.

Acknowledgements

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. FALCO's development was funded by the WFIRST Coronagraph Instrument (CGI) project and the NASA Exoplanet Exploration Program's (ExEP's) Segmented Coronagraph Design and Analysis (SCDA) study. The authors thank Dan Sirbu and Jorge Domingo Llop-Sayson for early testing and troubleshooting of FALCO.

REFERENCES

[1] Noecker, M. C., Zhao, F., Demers, R., Trauger, J., Guyon, O., and Jeremy Kasdin, N., "Coronagraph instrument for WFIRST-AFTA," *Journal of Astronomical Telescopes, Instruments, and Systems* **2**(1), 011001 (2016).

[2] Bolcar, M. R., Aloezos, S., et al., "The Large UV/Optical/Infrared Surveyor (LUVOIR): Decadal mission concept design update," in [*Proc. of SPIE*], **10398** (2018).

[3] Krist, J., "PROPER: an optical propagation library for IDL," in [*Proc. of SPIE*], **6675**, 66750P (2007).

[4] Perrin, M. D., Soummer, R., Elliott, E. M., Lallo, M. D., and Sivaramakrishnan, A., "Simulating point spread functions for the James Webb Space Telescope with WebbPSF," in [*Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave*], **8442**, 84423D (Sept. 2012).

[5] Por, E. H., Y. Haffert, S., M. Radhakrishnan, V., S. Doelman, D., van Kooten, M., and P. Bos, S., "High contrast imaging for python (hcipy): an open-source adaptive optics and coronagraph simulator," (2018).

[6] Ruane, G. J., Riggs, A. J. E., et al., "Review of high-contrast imaging systems for current and future ground- and space-based telescopes I. coronagraph design methods and optical performance metrics," in [*Proc. of SPIE*], **10698** (2018).

[7] Trauger, J., Moody, D., Krist, J., and Gordon, B., "Hybrid Lyot coronagraph for WFIRST-AFTA: coronagraph design and performance metrics," *Journal of Astronomical Telescopes, Instruments, and Systems* **2**(1), 011013 (2016).

[8] Sidick, E. et al., "Optimizing the regularization in broadband wavefront control algorithm for WFIRST coronagraph," in [*Proceedings of SPIE*], **10400**, 74 (2017).

[9] Pueyo, L. and Norman, C., "High-contrast imaging with an arbitrary aperture: Active compensation of aperture discontinuities," *Astrophysical Journal* **769**(102) (2013).

[10] Mazoyer, J., Pueyo, L., N'Diaye, M., Fogarty, K., Zimmerman, N., Leboulleux, L., Laurent, K. E. S., Soummer, R., Shaklan, S., and Norman, C., "Active Correction of Aperture Discontinuities-Optimized Stroke Minimization. I. a new adaptive interaction matrix algorithm," *ApJ* **155**(1), 7 (2018).

[11] Mazoyer, J., Pueyo, L., N'Diaye, M., Fogarty, K., Zimmerman, N., Soummer, R., Shaklan, S., and Norman, C., "Active Correction of Aperture Discontinuities-Optimized Stroke Minimization. II. optimization for future missions," *ApJ* **155**(1), 8 (2018).

[12] Give'on, A., Kern, B., Shaklan, S., Moody, D., and Pueyo, L., "Broadband wavefront correction algorithm for high-contrast imaging systems," *Proceedings of SPIE* **6691**, 66910A–1 – 66910A–11 (2007).

[13] Pueyo, L., Kay, J., Kasdin, N., Groff, T., McElwain, M., Give'on, A., and Belikov, R., "Optimal dark hole generation via two deformable mirrors with stroke minimization," *Applied Optics* **48**(32), 6296–6312 (2009).

[14] Sidick, E., Riggs, A. J. E., Ruane, G. J., Krist, J. E., Moody, D. C., and Coker, C. T., "Fast linearized coronagraph optimizer (FALCO) II: optical model validation and time savings over other methods," (2018).

[15] Coker, C. T., Ruane, G. J., Riggs, A. J. E., et al., "Fast Linearized Coronagraph Optimizer (FALCO) III. optimization of key coronagraph design parameters," in [*Proc. of SPIE*], **10698** (2018).

[16] Ruane, G., Riggs, A. J. E., Coker, C. T., et al., "Fast Linearized Coronagraph Optimizer (FALCO) IV. coronagraph design survey for obstructed and segmented apertures," in [*Proc. of SPIE*], **10698** (2018).

[17] Ruane, G. J. et al., "Laboratory testing of coronagraphs for future space telescopes on the Caltech High Contrast Spectroscopy Testbed for Segmented Telescopes," in [*Proc. of SPIE*], **10698** (2018).

[18] Bordé, P. J. and Traub, W. A., "High-contrast imaging from space: Speckle nulling in a low aberration regime.," *Astrophysical Journal* **638**, 488–498 (February 2006).

[19] Trauger, J. T., Moody, D., Gordon, B., Gursel, Y., Ealey, M. A., and Bagwell, R. B., "Performance of a precision high-density deformable mirror for extremely high contrast imaging astronomy from space," in [*Proc. of SPIE*], **4854**, 1–8 (2003).

[20] Bifano, T. G., Perreault, J. A., and Bierden, P. A., "Micromachined deformable mirror for optical wavefront compensation," (2000).

[21] Soummer, R., Pueyo, L., Sivaramakrishnan, A., and Vanderbei, R., "Fast computation of Lyot-style coronagraph propagation," *Optics Express* **15**(24), 15935–15951 (2007).

[22] Vanderbei, R., "Fast Fourier optimization: Sparsity matters," *Mathematical Programming Computation* , 1–17 (2012).

[23] Deng, X., Bihari, B., Gan, J., Zhao, F., and Chen, R. T., "Fast algorithm for chirp transforms with zooming-in ability and its applications," *J. Opt. Soc. Am. A* **17**, 762–771 (Apr 2000).

[24] Zimmerman, N. T., Riggs, A. J. E., Jeremy Kasdin, N., Carlotti, A., and Vanderbei, R. J., "Shaped pupil Lyot coronagraphs: high-contrast solutions for restricted focal planes," *Journal of Astronomical Telescopes, Instruments, and Systems* **2**(1), 011012 (2016).

[25] Juanola-Parramon, R., Zimmerman, N. T., et al., "Modelling exoplanet detection with the LUVOIR coronagraph," in [*Proc. of SPIE*], **10698** (2018).