Publications of the Astronomical Society of the Pacific, 131:024502 (7pp), 2019 February https://doi.org/10.1088/1538-3873/aaec2e


# Echelle++, a Fast Generic Spectrum Simulator

Julian Stürmer[1], Andreas Seifahrt[1], Zachary Robertson[1], Christian Schwab[2], and Jacob L. Bean[1]
[1] The Department of Astronomy and Astrophysics, University of Chicago, USA
[2] Department of Physics and Astronomy, Macquarie University, Australia
*Received 2018 September 11; accepted 2018 October 22; published 2018 December 13*

## Abstract

We present the software package, `Echelle++`, an open-source `C++` code to simulate realistic raw spectra based on the Zemax model of any spectrograph, with a particular emphasis on cross-dispersed Échelle spectrographs. `Echelle++` generates realistic spectra of astronomical and calibration sources, with accurate representation of optical aberrations, the shape of the point-spread function, detector characteristics, and photon noise. It produces high-fidelity spectra fast, a very important feature when testing data reduction pipelines with a large set of different input spectra, when making critical choices about order spacing in the design phase of the instrument, or while aligning the spectrograph during construction. `Echelle++` also works with low-resolution, low signal-to-noise, multi-object, IFU, or long-slit spectra, for simulating a wide array of spectrographs. We chose to initially generate our own spectrograph model from the optical prescription in Zemax. `Echelle++` can then be used independently, without access to commercial ray tracing software.

*Key words:* instrumentation: spectrographs – techniques: radial velocities – methods: numerical

*Online material:* color figures

## 1. Introduction

Échelle spectrographs are a vital tool for obtaining high-resolution spectra in a wide range of astronomical applications. Particularly for Doppler detection of exoplanets, modern Échelle spectrographs are used to measure radial velocities of stars with a precision of 1 m $s^{-1}$ or better (Fischer et al. 2016). Achieving this extreme precision requires both a very stable, well calibrated instrument, and a carefully designed data reduction and analysis software.

Having a data simulator that generates realistic stellar spectra is an important tool for the design of the instrument and the development of its data reduction and analysis software. In the design phase of an instrument, a simulator can be used for the verification of the optical design and required specifications. During assembly and integration, simulated spectra provide a convenient way of verifying optical alignment, and allow a direct comparison of model versus as-built optical performance. Even more importantly, testing the data reduction pipeline and analysis software using stellar spectra with a known radial velocity shift facilitates the development of a robust pipeline before commissioning the spectrograph, which saves time and mitigates risk. In the context of new reduction algorithms like the "perfect" extraction (Bolton & Schlegel 2010), a realistic simulation also allows for quantitative comparisons between a new code and established reduction methods.

To meet these needs, several instrument simulators have been developed in the past, such as the HERMES simulator (Goodwin et al. 2010), EchMod (Barnes 2012), the CARMENES Spectrum Forward Simulator[3] or the Zemax-based simulator by Gibson & Wishnow (2016). Unfortunately, only one of these simulators is open source, and most are written for a specific instrument or require concurrent access to a commercial optical design software, such as Zemax/OpticStudio. In addition, not all of the existing simulators handle aberrations accurately.

In the following, we present `Echelle++`, an open source, C++ based software package that provides a flexible simulator for generating realistic spectra, in particular for Échelles. The code can be found on our github repository.[4] For convenience, we also provide a docker image[5] that allows running `Echelle++` on all major platforms.

## 2. Motivation

A physical treatment of an Échelle spectrograph's optical elements as individual components is rather complex, and a full physical modeling of the entire optical system is cumbersome. For example, a physical model of the Échelle grating would in principle require the application of diffraction theory to a full 3D description of the grating surface. In practise, a number of approximations are required to allow a practical implementation of the physical model and to speed up computation time. While such an approach is possible and

[3] https://github.com/cjmarvin/csfs
[4] https://github.com/Stuermer/EchelleSimulator
[5] https://hub.docker.com/r/stuermer/echellesimulator/

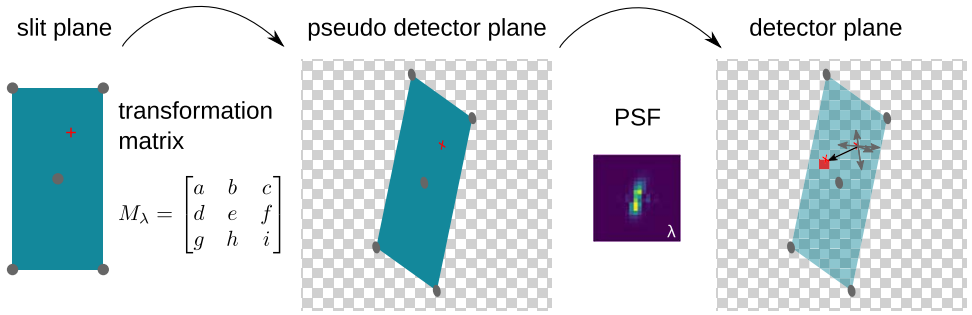


**Figure 1.** Illustration of the basic working principle of the simulator. A random point on the slit plane ($x$, $y$) of the spectrograph is selected (red cross) and mapped onto the detector plane ($x'$, $y'$) via a single transformation matrix ($M_\lambda$). The matrix is calculated by tracing the four corner points and the center of the slit aperture (gray dots) through the optical model (in Zemax). A wavelength-dependent PSF, also derived from the optical model, is used to account for optical aberrations by randomly disturbing the position of the traced photon ($x'$, $y'$) to calculate the final position of the traced point in the detector plane ($X$, $Y$). In the last step, the count value of the pixel hit by the photon is simply increased by one. Note that the actual form of the input slit does not need to be rectangular; the five points allow us to determine the affine transformation of the input focal plane.
(A color version of this figure is available in the online journal.)

has proved very helpful in verifying the optical design of certain instruments (e.g., Chanumolu et al. 2015, 2017), we argue that it is still too limited, too time consuming to adapt to different instruments, and too slow for the simulation of realistic 2D Échelle spectra.

Usually, commercial optical ray tracing software, like Zemax/OpticStudio, is employed for designing spectrographs, providing a very accurate physical model and a reasonably user-friendly interface to set up the optical model. In principle, the ray tracing software itself could be used in conjunction with additional code to simulate full 2D spectra. This approach was chosen by Gibson & Wishnow (2016). The advantage is a high level of sophistication in the optical model, incorporating, for example, diffraction effects or stray light, with minimal user input.

However, the Zemax/OpticStudio package is fairly slow for efficient 2D data simulation, requiring many hours of computation time for a full 2D spectrum on a 4k × 4k CCD (Steve Gibson, private communication). More importantly, commercial raytracting software is very costly and not readily accessible to all users interested in simulating Échelle data.

With these difficulties in mind, we developed `Echelle++`, a software package that uses spectrograph model files that are retrieved from Zemax once, so that no access to the optical design software is needed during the actual simulation. This approach is similar to the one presented by Goodwin et al. (2010) for the data simulator of the HERMES spectrograph.

## 3. Concept

The fundamental assumption of `Echelle++` is that the entire optical response of a spectrograph at any monochromatic wavelength can be described by a 3 × 3 transformation matrix applied to the input slit aperture, and a subsequent convolution with a point-spread function (PSF). Both the matrix and the PSF are wavelength dependent. It is reasonable to assume that across an order both will vary slowly, allowing us to calculate only a limited set of transformation matrices and PSFs and interpolate for any wavelength in between.

In a sense, a transformation matrix describes the geometric optical response of a spectrograph for a monochromatic wavelength as an affine transformation. The matrix maps any given location in the input slit onto the detector plane, accounting for both the wavelength-dependent location of the slit image in the detector plane as well as for geometric distortions of the slit image itself, e.g., slit rotation, shear, anamorphic magnification etc. Since we generate the matrices by tracing chief rays in the focal plane, higher order aberrations, effects on the spot shape, and diffraction effects are not incorporated in the transformation matrices. To also account for these effects, the calculation of a wavelength-dependent PSF and its application on the transformed image is necessary.

We choose to simulate the spectrum photon-by-photon, starting from a random position within the input slit for each photon drawn from an input spectrum, rather than transforming a number of complete monochromatic slit images. One distinct advantage of this approach is that no sampling effects are present in the resulting image, because the discretisation happens in the very last step, when the photons are binned into the detector pixels. Another benefit of this method is that the simulated spectrum intrinsically has the correct photon noise statistics. Figure 1 visualizes the simulation concept of `Echelle++`.

The only major downside is the very large number of photons needed for a high signal-to-noise (S/N) spectrum, making the simulation computationally expensive compared to a classical image transformation approach (see Section 4 for details). However, as tracing a single photon does not involve

the solution of an optical equation at each optical surface in the spectrograph, but rather the interpolation of a limited number of matrix elements, it is much faster than ray tracing in Zemax/OpticStudio itself.

## 4. Implementation

`Echelle++` is essentially a ray-tracing simulator with two distinct steps. In step one, a spectrograph model with discrete transformation matrices and PSFs is constructed using Zemax as described in Sections 4.1 and 4.2 below. In step 2, which can now be executed independent of Zemax, 2D spectra are simulated based on this simplified recipe:

1. Select a random position $(x, y)$ in the slit or fiber aperture.
2. Select a random wavelength $\lambda$ for a given source spectrum $F(\lambda)$ & instrument transmission curve $T(\lambda)$.
3. Calculate the interpolated transformation matrix $M_\lambda$.
4. Calculate the chief ray position in the detector plane $(x', y') = M_\lambda \cdot (x, y)$
5. Calculate the interpolated point-spread function $\mathrm{PSF}_\lambda$.
6. Apply a random perturbation of $(x', y')$ based on the probability distribution $\mathrm{PSF}_\lambda$ to get the final position in the detector plane $(X, Y)$.
7. Increase the pixel value of the nearest pixel to $(X, Y)$ by one.

These steps are repeated $N_\nu$ times, where $N_\nu$ is the total number of photons in the simulation. This number is usually very large, depending on the desired signal to noise (S/N) ratio of the spectrum. For example, to simulate a S/N 200 spectrum, (where the S/N is measured per resolution element) on the order of $5 \cdot 10^7$ photons per order need to be traced when assuming 4000 pixels in dispersion direction and 3 pixel sampling.

### 4.1. Transformation Matrices

In principle, there are two geometric transformations we could choose from: affine and projective transformation. While the latter is more general, we found that affine transformations are sufficiently accurate to describe the spectrograph optics. Affine transformation parameters also give an intuitive insight in what happens across an order, because they can be expressed in terms of rotation ($\Theta$), shear ($\xi$), scaling in both directions ($s_x$ and $s_y$) and translation ($t_x$ and $t_y$). There are various methods to compose an affine transformation matrix; we use the following:

$$M_\lambda = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & m_{22} \end{bmatrix} \overset{\text{(affine)}}{=} \begin{bmatrix} s_x \cdot \cos(\Theta) & -s_y \cdot \sin(\Theta + \xi) & t_x \\ s_x \cdot \sin(\Theta) & +s_y \cdot \cos(\Theta + \xi) & t_y \\ 0 & 0 & 1 \end{bmatrix}.$$

We choose to calculate the wavelength-dependent transformation parameters by tracing the four corner points as well as the center of the input field (i.e., fiber/slit) through Zemax, obtaining the coordinates of these five points in the detector plane and applying a set of standard algebraic equations to calculate the transformation matrices for a number of wavelength steps (~50–100) per Échelle order. As an example, Figure 2 shows the matrix parameters for MAROON-X, a radial velocity spectrograph for the Gemini Observatory (Seifahrt et al. 2018), for two different Échelle orders as a function of pixel coordinate along the order.

It is worth taking a closer look at the matrix parameters, in particular the ones decomposed in the canonical transformations (Figure 2).

The coordinate systems of the input and the detector plane can be chosen arbitrarily during the definition of the transformation matrices, and we decided to use normalized coordinates in the input slit and pixel coordinates for the detector plane. Using these coordinates, the scaling parameters ($s_x$ and $s_y$) directly show the sampling of the bounding box of the slit in pixels in dispersion and cross-dispersion direction, respectively. For MAROON-X, the anamorphic magnification caused by the Échelle grating across an order can easily be seen in those plots. Note that these numbers are lower limits for the pixel sampling, as the matrices do not account for optical aberrations and therefore do not include the additional "blur" by the PSF, which will be applied in a later step.

As an additional benefit, the resulting spectrograph model is well suited to be used directly in the data reduction software since the model provides good estimates for the position and labels of individual fibers and orders on the echellogram. It also provides a good initial guess for the wavelength solution. This feature has been incorporated in the data reduction pipeline for MAROON-X.

### 4.2. Point-spread Functions

As the transformation matrices are calculated from ray tracing of chief rays only, they do not contain the full information of the optical aberrations of a spectrograph at a certain wavelength. However, these aberrations are important for a realistic simulation as they contribute considerably to the line shape in a spectrograph. For most spectrographs, the shape of the PSF cannot be described analytically. The PSF is usually a complex function that results from the interplay of various optical aberrations like coma, astigmatism, and spherical aberration as well as diffraction patterns. Therefore, we use the Huygens PSF functionality of Zemax to calculate 2D PSFs at discrete positions along each order, which are later interpolated for a given wavelength. The resulting local PSF shape is then interpreted as probability density for the relative

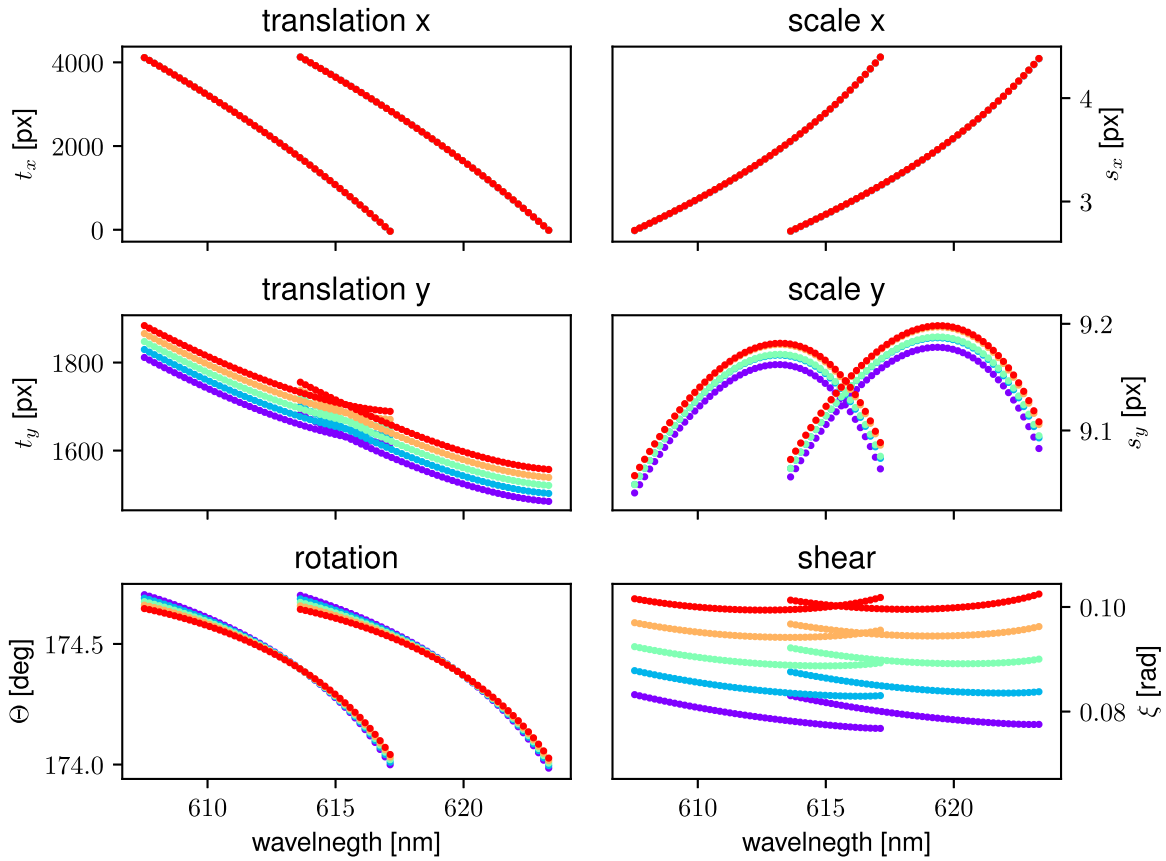


**Figure 2.** Matrix parameters of MAROON-X for orders 99 and 100 decomposed into geometric transformations. The five different colors indicate the five different fibers in the entrance slit of the spectrograph, hence different positions in the virtual entrance slit, shifted in cross-dispersion direction. All parameters vary smoothly across a single order which allows for interpolation at intermediate wavelengths.
(A color version of this figure is available in the online journal.)

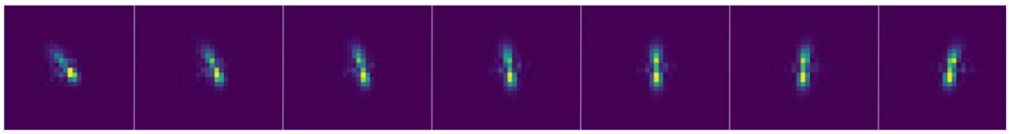

**Figure 3.** Across each order, a number of PSFs are calculated that are used as model PSFs for the given wavelength. Here, the central seven PSFs of the central fiber of order 100 of MAROON-X are shown. One box is approximately the size of one detector pixel. The PSFs are linearly interpolated for intermediate wavelengths and used as 2D probability density functions for a random displacement that is applied in the simulation process for a single photon.
(A color version of this figure is available in the online journal.)

displacement of a single photon with respect to the chief ray coordinate traced into the detector plane by the application of the transformation matrix. Figure 3 shows typical PSFs of MAROON-X as an example for a multimode fiber-fed spectrograph.

### *4.3. Spectra and Efficiency Curves*

While the transformation matrices and the PSFs handle the correct location for each photon, 1D input spectra and efficiency curves are needed to correctly model the intensity distribution in the 2D spectrum. A continuous 1D spectrum defined at discrete points (for example, a stellar or a flat-field spectrum), as well as a list of wavelengths (like a list of ThAr lines or peak positions of a Fabry–Perot etalon or laser frequency comb) can be used to define an intensity distribution for the selection of wavelengths in the randomization process.

`Echelle++` supports .csv and .fits files which provide a discrete spectrum or a line list.

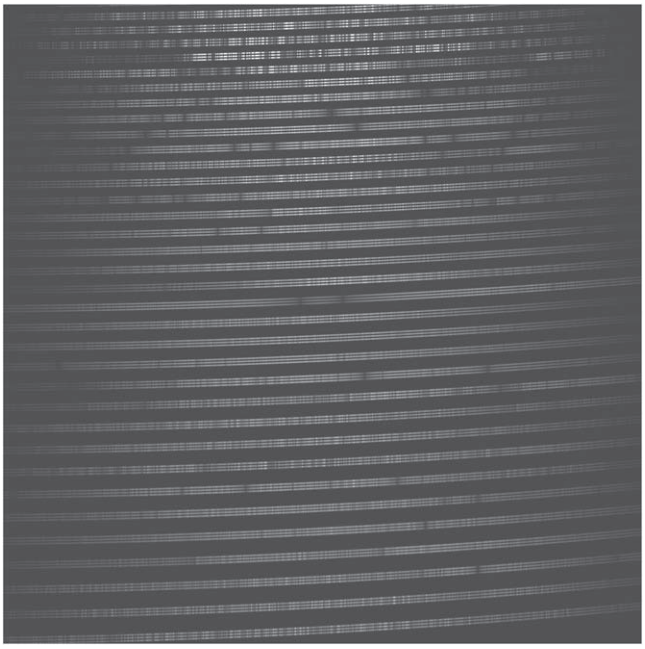

**Figure 4.** Full-frame simulated M-dwarf spectrum using the MaroonX spectrograph model.

For convenience, we implemented the direct generation of a set of commonly used spectra, namely a constant flat spectrum, a blackbody spectrum, model M-dwarf spectra based on PHOENIX (Husser et al. 2013) and support for Sun-like Coehlo spectra (Levenhagen et al. 2017). For example, in order to simulate a PHOENIX spectrum, it is sufficient to specify the characteristic stellar parameters and Echelle++ will download the appropriate PHOENIX file (see also Section 5). For celestial sources, the V-band magnitude and the telescope aperture in addition to the exposure time should be specified to obtain the correct flux. Combined with the instrument's efficiency model, the echellograms generated by Echelle++ have the correct S/N ratio.

To account for the wavelength-dependent transmission of the instrument itself, Echelle++ permits modification of the source spectrum through multiplication with efficiency spectra describing various effects. These additional spectra are interpolated onto the wavelength grid of the underlying source spectrum and multiplied with the latter. Effects that shape the transmission spectrum include the blaze function of the Échelle grating (defined for each Échelle order), the measured or modeled efficiency of mirrors, glasses, and coatings, or the atmospheric transmission above the observatory. Telluric contamination of the spectra can also be included as an efficiency spectrum; we are providing an example of how to include an atmospheric telluric spectrum to be used as an efficiency file. The code can handle multiple efficiency spectra simultaneously.

The efficiency curves are highly dependent on the specific instrument. For convenience, a function is implemented that provides a good approximation of the blaze function (Casini & Nelson 2014). By default, the necessary parameters are read from the spectrograph model file, and the normalized blaze function is calculated and applied to the input spectrum. All other efficiency curves have to be provided by the user in form of a .csv, .hdf, or .fits file.

After applying the wavelength-dependent instrument efficiency to the input spectrum, Echelle++ calculates the spectral flux distribution in photons per second. In the case of a continuous input spectrum (as opposed to a discrete emission line spectrum), a cumulative density distribution (CDF) per order is calculated. For a given integration time, the resulting number of wavelengths are drawn from the CDF using inverse-transform sampling. In case of a line list as input spectrum, the wavelengths are drawn from the discrete set of lines instead. To guarantee good randomness, we use a Marsenne twister (Matsumoto & Nishimura 1998) as pseudo-random generator.

## 5. How to use Echelle++

In order to use Echelle++, a spectrograph model file has to be created from the Zemax optical design via a python script. Since there is no standard among optical designers on how to set up the spectrograph design in Zemax, it is difficult to write a generic script. Therefore, it usually takes minor manual tweaking of the Zemax file in order to generate the spectrograph model. Echelle++ uses the comment column of the Zemax file to identify the grating's surface number, as well as the incidence- and the gamma angle of the grating. It simply requires these parameters to be in separate rows with unique names in the comment field. If the incidence angle is not identical to the blaze angle (that is, if the grating is not mounted in Littrow configuration), the blaze angle of the grating can be specified separately. In case of a low-resolution spectrograph which uses only one order of the grating, or a prism, the model file can be created without these parameters. If the grating angles and the order numbers are provided, Echelle++ calculates a normalized blaze function to estimate the grating's efficiency.

We provide a working example Zemax file (along with the model creation script) that shows how the spectrograph could be set up.

The authors are available to assist in adjusting a Zemax file to the needs of Echelle++. The MAROON-X, VeloceRosso, HPF and NEID spectrographs are already supported. We are interested in adding more models and encourage spectrograph teams to contact us.

Once the spectrograph model is available and the code is successfully compiled, there are two ways of using Echelle++.

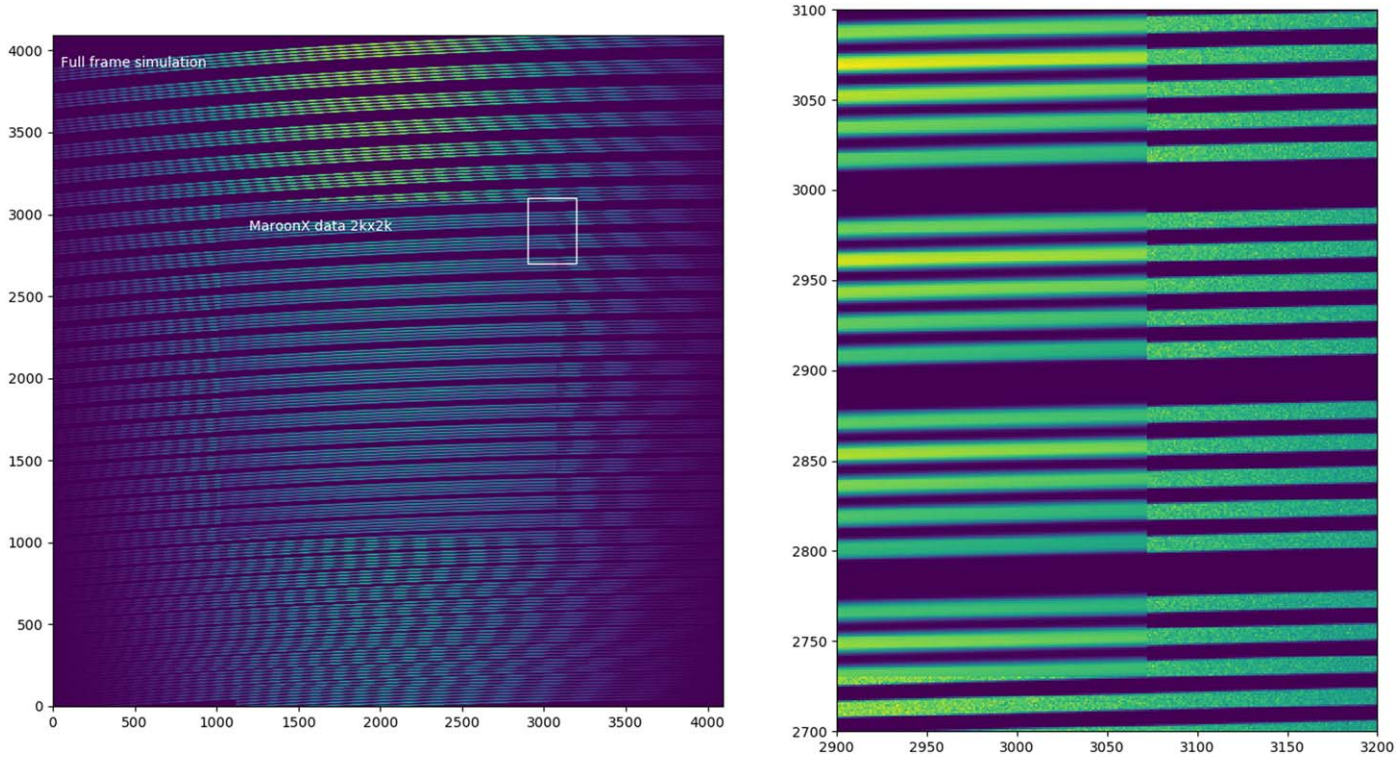


**Figure 5.** Comparison between a MAROON-X simulation of a flat-field spectrum and an actual measurement taken during initial alignment of the spectrograph. On the left side, we show the simulation for the 4k × 4k detector while the inset shows the actual measurement taken with an interim 2k × 2k detector used for preliminary testing. On the right side, a zoomed-in view of the border between simulated and real spectra. Note that for this comparison, we did not further adjust the Zemax model to the is-built system.
(A color version of this figure is available in the online journal.)

The first way is using the provided executable, which is controlled via command line arguments. The following command lists all available arguments and their purpose:

```
$ ./Echellesimulator -h
```

For example,

```
$ ./Echellesimulator -s MaroonX -f 1 --phoenix ↪ 3200, -1.,0.,-5.5,1 -t 100 --radial-velocity 55 -o ↪ mdwarf.fit
```

simulates an M-dwarf model spectrum in the first fiber of the MAROON-X spectrograph. It downloads a `PHOENIX` model with the given stellar parameters and magnitude 1, applies a radial velocity shift of 55 m $s^{-1}$, and simulates a 100 s integration time. The output is saved as the FITS file mdwarf.fit. The spectrum, after repeating the simulation for two more fibers, is shown in Figure 4. As a second example, Figure 5 illustrates a good match for the MAROON-X spectrograph between simulated data and a real measurement.

Because the whole program is controlled via command line arguments it is easy to write scripts to generate, e.g., a radial velocity sequence of the same target. Examples of how to script `Echelle++` via python can be found in the examples folder of the repository.

The second way to use `Echelle++` is writing a custom application that uses the `C++` classes and functions directly. This approach is meant for cases where heavy customization is required; for most cases, the command line access is the recommended way to use it.

On a current laptop (Intel Xeon E3-1545M, 4 CPU cores), `Echelle++` reaches a simulation speed of about $10^7$ photons per second. This results in a processing time of about 5 s per order for a S/N 200 spectrum (4096 pixels, 3 pixel sampling). The program uses OpenMP[6] for multiprocessing, and the processing time decreases proportionally to the number of cores, plus some minor overhead.

## 6. Conclusion and Outlook

Currently, `Echelle++` is able to simulate realistic 2D spectra for any slit or fiber-fed spectrograph. It uses spectrograph models that are extracted directly from Zemax resulting in a model that accurately describes the dispersion and

[6] http://www.openmp.org/

optical aberrations. To run the actual simulations, no access to Zemax is required.

Any model spectrum that is provided by the user in .csv or .fits format can be simulated. For convenience, we have implemented internal functionality to simulate several common types of spectra based on parameters or publicly available standard spectra (flats, blackbodys, M-dwarfs, Sun-like stars, etalons, laser frequency combs, hollow cathode lamps).

If the spectrograph input is comprised of a number of individual fibers, each 2D spectrum from a single fiber is simulated separately, either using the same or a different source spectrum, and can be simply co-added into a final frame. Finally, CCD bias offsets and read noise can be added. While there is no in-built function yet to calculate and add sky emission and telluric absorption features, they can simply be treated as additional source spectra (in the case of sky emission) or as transmission data (in the case of telluric absorption).

The photon-wise simulation of the spectra is ideally suited for massive parallelization. Because most time of the simulation is spent in generating random numbers and simple matrix multiplications, the authors estimate that by partially transferring the calculations to a GPU, a 10–100x speedup should be achievable.

One of the main motivations for developing `Echelle++` was to test and optimize data reduction pipelines. Achieving the above stated speedup would allow `Echelle++` to be also used as a data reduction tool. Forward modeled spectra could be directly compared and matched to real echellograms by slightly perturbing the spectrograph model and the stellar spectrum (i.e., for measuring radial velocity shifts).

The concept of `Echelle++` also allows the addition of features in the future which are important for simulating effects that are relevant when pushing spectrographs and data analysis codes well beyond the current precision limit. For example, changes in the spatial illumination pattern, either from insufficient spatial scrambling or modal noise in the fibers, can be simulated by incorporating a specific spatial probability function for the slit image. In the current implementation, positions in the entrance slit aperture are chosen randomly from a uniform distribution of either a rectangular or a circular area. Replacing the uniform distribution with a different, even wavelength-dependent one, is possible. This allows the simulation of long-slit and IFU spectra which preserve spatial information within the slit.

Another important problem which is often ignored in simulating spectra are detector effects, such as intrapixel variations of quantum efficiency, variable pixel sizes and offsets (i.e., stitching errors), or artifacts arising from imperfect charge transfer efficiency or the brighter–fatter effect found in thick high-resistivity bulk silicon detectors. These could be simulated as final add-ons in the last step of the simulation, when the physical coordinates of each simulated photon are mapped onto a discrete pixel grid.

## References

Barnes, S. I. 2012, Proc. SPIE, 8550, 85501T
Bolton, A., & Schlegel, D. 2010, PASP, 122, 248
Casini, R., & Nelson, P. G. 2014, JOSSA, 31, 2179
Chanumolu, A., Jones, D., & Thirupathi, S. 2015, ExA, 39, 423
Chanumolu, A., Thirupathi, S., Jones, D., et al. 2017, ExA, 43, 39
Fischer, D. A., Anglada-Escude, G., Arriagada, P., et al. 2016, PASP, 128, 066001
Gibson, S. R., & Wishnow, E. H. 2016, Proc. SPIE, 9911, 99112D
Goodwin, M., Smedley, S., Barnes, S., et al. 2010, Proc. SPIE, 7735, 77357U
Husser, T.-O., von Berg, S. W., Dreizler, S., et al. 2013, A&A, 553, A6
Levenhagen, R. S., Diaz, M. P., Coelho, P. R. T., & Hubeny, I. 2017, ApJS, 231, 1
Matsumoto, M., & Nishimura, T. 1998, ACM Trans. Model. Comput. Simul., 8, 3
Seifahrt, A., Stürmer, J., Bean, J. L., & Schwab, C. 2018, arXiv:1805.09276