Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **475,** 4170–4185 (2018)
Advance Access publication 2018 January 12

doi:10.1093/mnras/sty101

COCOA code for creating mock observations of star cluster models

Abbas Askar,[1]★ Mirek Giersz,[1] Wojciech Pych[1] and Emanuele Dalessandro[2,3]

[1]*Nicolaus Copernicus Astronomical Centre, Polish Academy of Sciences, ul. Bartycka 18, PL-00-716 Warsaw, Poland*
[2]*INAF – Osservatorio Astronomico di Bologna, Via Gobetti 93/3, I-40129 Bologna, Italy*
[3]*Dipartimento di Fisica e Astronomia, Università degli Studi di Bologna, Via Gobetti 93/2, I-40129 Bologna, Italy*

Accepted 2018 January 8. Received 2018 January 4; in original form 2017 March 24

ABSTRACT

We introduce and present results from the COCOA (Cluster simulatiOn Comparison with ObservAtions) code that has been developed to create idealized mock photometric observations using results from numerical simulations of star cluster evolution. COCOA is able to present the output of realistic numerical simulations of star clusters carried out using Monte Carlo or *N*-body codes in a way that is useful for direct comparison with photometric observations. In this paper, we describe the COCOA code and demonstrate its different applications by utilizing globular cluster (GC) models simulated with the MOCCA (MOnte Carlo Cluster simulAtor) code. COCOA is used to synthetically observe these different GC models with optical telescopes, perform point spread function photometry, and subsequently produce observed colour–magnitude diagrams. We also use COCOA to compare the results from synthetic observations of a cluster model that has the same age and metallicity as the Galactic GC NGC 2808 with observations of the same cluster carried out with a 2.2 m optical telescope. We find that COCOA can effectively simulate realistic observations and recover photometric data. COCOA has numerous scientific applications that maybe be helpful for both theoreticians and observers that work on star clusters. Plans for further improving and developing the code are also discussed in this paper.

Key words: methods: numerical – methods: observational – techniques: photometric – globular clusters: general – galaxies: star clusters: general.

1 INTRODUCTION

Because of advancements in computational technology and programming over the past couple of decades, there has been extensive work done in modelling the dynamical evolution of star clusters using sophisticated numerical simulation codes like direct *N*-body and Monte Carlo codes. The long-standing goal of simulating the evolution of a realistic million-body globular cluster (GC) up to 12 billion yr with a direct *N*-body code was recently achieved by Wang et al. (2016) with the NBODY6++GPU code (Wang et al. 2015). With increasingly more realistic simulations of star clusters, there is also a need to validate and compare results with observations. For this purpose, it is important to present simulation results in a way that would be most useful for direct comparison with observations of star clusters. In this paper, we introduce and present the COCOA (Cluster simulatiOn Comparison with ObservAtions) code[1] that can automatically simulate mock photometric observations of star clusters from simulation snapshots and then subsequently apply observational methods and techniques to obtain cluster parameters from the mock images. COCOA will allow theoreticians working with numerical simulations to output their simulated cluster model data as a realistic observation enabling them to get feedback from observers and vice versa. COCOA can create synthetic ideal observations from virtually any optical telescope and imaging camera and has an inbuilt point spread function (PSF) photometry pipeline that can perform photometry on the simulated images and obtain star catalogues by utilizing procedures that are typically employed by observers. COCOA has already been used to create mock observations of *N*-body and Monte Carlo models after 12 Gyr of cluster evolution in Askar et al. (2016, 2017a), Wang et al. (2016), and Belloni et al. (2017).

Numerical simulation codes provide detailed output showing the evolution of global parameters of cluster models and also snapshots containing relevant information for all objects in the star cluster at a specific time. Such detailed output can be extended so that results from simulation models of star clusters can be presented in a format that can be utilized and understood by observers studying objects like GCs. This will not only facilitate a direct comparison between simulated models and observed clusters but will also be effective in determining the accuracy of observational techniques and methods to determine different cluster parameters. The importance of this endeavour to bridge the gap between theoreticians and observers working on star clusters was recognized by Hut et al. (2002) and efforts were made to simulate synthetic observations of star

★ E-mail: askar@camk.edu.pl
[1] https://github.com/abs2k12/COCOA

cluster simulations within the STARLAB environment (Hut et al. 2010). Stellar and binary evolution routines (Hurley, Pols & Tout 2000; Hurley, Tout & Pols 2002) implemented within NBODY6 and other cluster evolution codes allowed the possibility to obtain magnitudes in few optical filters that could be used to create colour–magnitude diagrams (CMDs) from simulation snapshots. Various studies using numerical simulation codes to investigate dynamical evolution and stellar populations in star clusters have employed different methods to compare their results with observations. For instance, Lamers, Anders & de Grijs (2006) used the GALEV code with *N*-body results to investigate the photometric evolution of dissolving clusters, Sippel et al. (2012) used projected *N*-body snapshots and created FITS images of model GCs to measure their half-light radii, and Zhuang et al. (2015) computed integrated colours and Lick indices for *N*-body models of GCs. Also more recently, Bianchini et al. (2015) have developed the SISCO code to simulate integral field unit (IFU) observations of GCs using data from numerical simulations that opens up the possibility to obtain observationally resolved kinematic measurements of stars in simulated GC models.

An important step towards comparing observations with simulations was the development of the MASSCLEAN code (Popescu & Hanson 2009, 2010, 2014) that could simulate the evolution of star clusters with different initial conditions and create synthetic observational images for those clusters. MASSCLEAN uses stellar evolution tracks to evolve the stars in the cluster model and for dynamics gives the user the option to enable simple linear mass segregation. Although the MASSCLEAN code does not simulate detailed dynamical evolution of the cluster and does not contain binary systems, it can output cluster data in the form of observational images and synthetic CMDs that can be compared with observations. By developing COCOA we seek to continue this work and allow users the possibility to use the results of realistic dynamical simulations from *N*-body and Monte Carlo codes to create observational photometric data that can be automatically reduced with the PSF photometry pipeline provided within the code.

With COCOA it is possible to check if there are any systematics or biases associated with actual observational data and techniques used to determine cluster properties. Our idealized mock observations do not include foreground and background stars, the noise is limited to a Poisson noise, there are no effects of cosmic rays on our images, and the PSF does not vary. Actual observations are plagued by all the aforementioned problems. Guidi, Scannapieco & Walcher (2015) have compared the direct results of galaxy simulations with the observationally derived quantities from synthetic observations and found that there are systematic differences in obtained galaxy parameters that are in part caused by observational biases. Sollima et al. (2015) have also shown that fitting analytic models to available observables for star clusters can result in significant biases while determining global parameters such as mass of the cluster. With COCOA it is possible to investigate such biases and systematics in photometric studies of star clusters. For instance, the influence of photometric errors on determination of cluster parameters like binary fractions can be checked. Accuracy of techniques used to obtain the centre of observed GCs can also be tested with COCOA. The completeness of observations can be checked by comparing the results from star catalogues obtained after doing photometry on the simulated images of clusters with the snapshot data that includes information for all stars in the cluster. Moreover, simulating such ideal observations from simulation models will not only provide a critical check on the data reduction processes and techniques used by observers but it will also be helpful in refining initial conditions for simulated cluster models aiming to reproduce Galactic GCs.

Furthermore, COCOA can also be used by observers for the purposes of writing observational proposals based on mock observations of results from realistic numerical simulation models of star clusters. Uncertainties as to how simulated cluster models would appear if they were observed as a real cluster on the sky can also be checked. Observational analysis of simulated cluster models may reveal numerical issues and problems with erroneous initial conditions of simulated models. As mentioned earlier, a thorough comparison between simulated models and observations can help better constrain initial model, parameters that control stellar evolution, binary parameter distribution, mass functions, and other initial structural assumptions that are used by theorists. This makes COCOA an important tool that may help observers in assisting theoreticians to improve their theoretical models.

In this paper we will mainly concentrate on describing COCOA and using its basic features for creating mock observations of results from MOCCA code (Giersz et al. 2013, and reference therein) for star cluster simulations and generating observational CMDs after carrying out PSF photometry on mock images. More sophisticated COCOA applications for generating surface brightness profiles, velocity dispersion profiles, number density profiles, luminosity functions, and fitting them to different models will be discussed in future papers. COCOA has been primarily developed in PYTHON and it combines various tools in other programming languages to offer different functionalities. In the second section of this paper, we provide a description of the COCOA code and how it works. In the third section we show results from COCOA by creating mock observations of cluster models that were simulated using the MOCCA code.

In the same section, we also present comparisons with an observation of the Galactic GC NGC 2808 and other different applications of the COCOA code. In the fourth section we discuss future developments and modules that will be developed for applying various techniques to the synthetic observational data. An appendix with a brief manual for the code is also provided.

2 CREATING MOCK OBSERVATIONS OF STAR CLUSTERS

COCOA was designed in a way that would be most user-friendly and with as many automated procedures as possible to assist users that are not familiar with observational reduction procedures. COCOA comes with default values and a working example that shows how to use the different features of the code. The COCOA code has essentially three modules. The first one is the projection module that projects numerical snapshot data for all stars in the cluster model on to the plane of the sky. The procedure and features of the projection module are provided in Section 2.1. The second module simulates the observation using the projected snapshot. The user provides the module with the parameters of the instrument/telescope with which they want to simulate the observations and it automatically generates the required number of frames to create a composite image of the entire cluster. The details of this module are provided in Sections 2.2. The third module is the automated PSF photometry pipeline in COCOA that carries out photometry on the simulated images produced using second module and the output from the PSF photometry code is used to create a catalogue of all observed stars in the cluster. The details of this module are provided in Section 2.3. A flow diagram illustrating the main steps of the code is shown in Fig. 1.

The simulation results used to generate mock observations in this paper come from the MOCCA code for star cluster simulations. MOCCA is based on the orbit averaged Monte Carlo method of Hénon (1971) for following the long-term evolution of spherically symmetric star

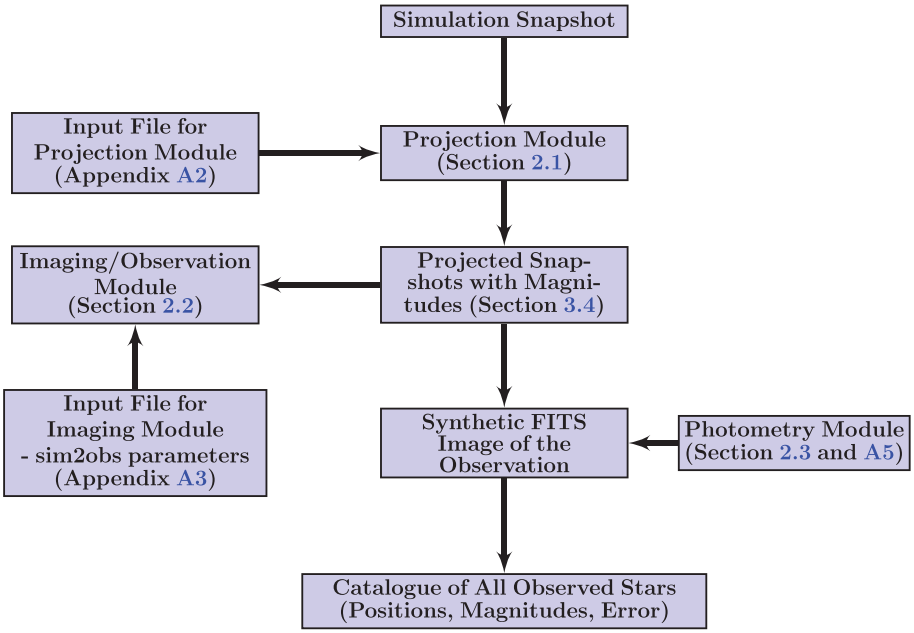

Figure 1. COCOA flow chart: a schematic diagram of the main steps and the sequence of the code.

clusters that was further improved by Stodolkiewicz (1986) and Giersz et al. (2013). Monte Carlo codes for star cluster evolution are more approximate than direct *N*-body codes but they are significantly faster. MOCCA uses the FEWBODY code (Fregeau et al. 2004) to compute strong binary–single and binary–binary interactions. For stellar and binary evolution, MOCCA uses SSE and BSE (Hurley et al. 2000, 2002). MOCCA was recently used to carry out a survey of about 2000 star cluster models, MOCCA Survey Database I (Askar et al. 2016) that samples a wide array of initial parameters that are important in determining the evolution of cluster models. Results from MOCCA have been extensively compared with *N*-body simulations and the agreement for both the evolution of global parameters and number of specific objects and binary systems is remarkable (Giersz et al. 2013; Wang et al. 2016; Madrid et al. 2017). For using COCOA, one needs two-dimensional positions and absolute magnitudes of stars that could be provided by either Monte Carlo or *N*-body codes.

2.1 Projecting simulation snapshot

As an input, COCOA requires a snapshot at any given time from numerical star cluster simulations. The results presented in this paper utilize the snapshots from the MOCCA code. These snapshots from MOCCA simulations are produced periodically during the cluster evolution and contain the details of all the objects in the GC model at a given time, these include their spatial, kinematic, and stellar evolution properties [provided by the SSE (Hurley et al. 2000) and BSE (Hurley et al. 2002) codes used by MOCCA] like position from the centre of the cluster, velocities, mass, radius, bolometric luminosity, magnitudes in three filters (*B*, *V*, and *I*), and other important parameters like semimajor axis and eccentricity for binaries. COCOA can project the positions and velocities of the star on the plane of the sky and decompose the positions and velocities of individual stars in binary systems. The procedure for decomposing the binaries uses the semimajor axis of the relative orbit, its eccentricity and masses of stars to compute semimajor axes of the barycentric orbits of the individual stars. The projected snapshot that COCOA outputs automatically provides additional parameters like the Keplerian orbital elements for binary systems that are taken from a random distribution functions. Using one simulation snapshot it is also possible to use COCOA to simulate additional projected snapshots on shorter user-defined time-scales (hours, days to years) in which the position and velocities of individual stars in binaries can be tracked by COCOA. The projection module can also be used to rotate and change the orientation of the observed cluster model. This can be very valuable in understanding the influence of how viewing a star cluster at a particular orientation influences the derived observational parameters.

2.2 Imaging cluster models

Using two-dimensional (2D) positions from our projected snapshot, the SIM2OBS[2] tool in COCOA is utilized to generate FITS images of simulated cluster models. The SIM2OBS tool requires an input parameter file and the projected snapshot (from the projected snapshot, SIM2OBS requires the *x*, *y* position of each star and its absolute magnitude) to run. The input parameter file for SIM2OBS requires a number of parameters that primarily define the properties of the mock observation, particularly the telescope, the imaging camera, the PSF model, and distance to the cluster model. The properties for the instrument and the imaging camera include the size of the image, exposure time, saturation level, gain, etc. A full list and explanation of these parameters is provided in Appendix A3. Using these user entered parameters SIM2OBS computes the flux from each star and the corresponding number of counts to accurately produce synthetic FITS images of the star cluster model. Another important parameter that SIM2OBS requires is image offset values from the centre of the cluster. When offset values are zero then the centre of the synthetic mock observation is always the cluster centre. By changing the offset values, it is possible to point towards a specific part of the cluster in order to observe it.

[2] Developed for COCOA by Wojciech Pych in the programming language C. Can also be used as a stand-alone tool (see Appendix A3).

The SIM2OBS tool can be used to image an entire star cluster model up to its effective radius by creating multiple mock observations by changing the offset parameter in SIM2OBS. A procedure has been devised in COCOA to automatically create multiple frames that collectively image the entire cluster model as a mosaic. When projecting the simulation snapshot, the code determines the effective radius of the cluster that is then converted to arcseconds using the user entered distance to cluster. Using the pixel scale value and the image size provided by the user, the code can determine the total number of frames that will be required to image the cluster from the centre up to the limiting radius. An overlap factor can also be added by the user to make sure stars at the edge of the frame are accounted for in the neighbouring frame. Offset values are determined and multiple parameter files are written for SIM2OBS that are then run through a script to create FITS files for different frames that when combined image the entire cluster. The FITS image files can be viewed with astronomical image viewing applications such as SAOIMAGE DS9, CASA VIEWER, or STARLINK GAIA. These applications can also be used to export the FITS image in commonly used picture formats.

2.3 Automated PSF photometry

Once FITS images of different frames that collectively observe the entire cluster star cluster model have been created (see Appendix A3 for more details of this procedure), it is then possible to carry out photometry on these files to recover positions and magnitudes of stars in the same way as observers would do. In order to do this, a PSF photometry pipeline that uses DAOPHOT/ALLSTAR (Stetson 1987, 1994) stand-alone program has been developed in COCOA. The automated photometry pipeline in COCOA makes use of wrappers to call DAOPHOT/ALLSTAR. This approach is, in part, similar to the photometry pipeline developed by the SMARTS XRB team (Buxton et al. 2012) for carrying out photometry of optical and infrared observations of X-ray binaries. The following steps are taken to perform the PSF photometry.

(i) The photometry module goes through each FITS file of an observed frame one by one. The first step is to determine the full width at half-maximum (FWHM) value of the PSF from the mock observation of the image. This value of the FWHM in the simulated images is given by the ratio between the seeing and the pixel scale value that gives the spatial resolution of the mock observation (see Appendix A3). This value can be directly provided to the module or one may use a stand-alone program[3] that is provided with COCOA to obtain the FWHM of the FITS file.

(ii) The value for the FWHM is then used to prepare the option files required by DAOPHOT and ALLSTAR to carry out PSF photometry. The value of the FWHM and the values entered by the user while generating the synthetic observations are used to determine the parameters required in the options file. These parameters and how they depend on the FWHM are given in Appendix A5.

(iii) After the option files to prepare DAOPHOT and ALLSTAR are written, the photometry module creates a wrapper for DAOPHOT by writing a BASH script that will automatically run the code. The wrapper is called with the image name, this allows DAOPHOT to know which file to process. Using the standard commands in DAOPHOT, we obtain a PSF model and all the required output files from DAOPHOT that are needed to do our PSF photometry. These standard commands involve attaching the image file and then carrying out aperture photometry on the image. The *pick* command in DAOPHOT is then used to identify stars to create the PSF model. The number of stars to be picked and their limiting instrumental magnitude is provided in the wrapper. Three iterations of the *psf* command in DAOPHOT are used to get a reliable PSF model.

(iv) Once the PSF model is generated, another wrapper is created to run ALLSTAR via a BASH script. The required input file names are provided and then the script is executed. The ALLSTAR code runs and outputs the final catalogue. The COCOA pipeline reads the contents of the catalogue file produced by ALLSTAR and stores the star number, x coordinate of the position, y coordinate of the position, magnitude, and error in the magnitude in an array. The positions in the catalogues produced by ALLSTAR are in the units of pixels. The code automatically converts the positions to arcsecond (using the pixel scale value entered by the user when FITS images were being generated) and then using the offset values it corrects the x and y positions in arcseconds so that we have the positions with respect to the centre of cluster. The corrected positions, magnitudes, and error in magnitude are written to an output file.

(v) The code then moves on to the next FITS image and repeats the steps 1–4 and the catalogue from each FITS image is appended to our output file.

(vi) Once all DAOPHOT and ALLSTAR have run on all the frames that image the entire cluster, the code then runs the catalogue through a script that cleans out the overlapping stars. This script compares position of all stars with each other and if there are duplicate stars in the catalogue that came from two different frames then for simplicity, the one with the lower photometric error is selected. After cleaning for overlap stars, we get a final clean catalogue of all stars in the GC.

2.3.1 Magnitude corrections and using single PSF models for multiple frames

It is required to convert instrumental magnitudes to apparent or absolute magnitudes for meaningful comparisons. In order to do this, we prepare a list of around a hundred well-separated single stars with a range of different absolute magnitudes. We created a mock observation for these hundred stars by using the exact same parameter files that were used to image the cluster models. The stars are placed at the same distance as the cluster model and have been given artificial positions that form a grid structure on the projected plane. The stars are well separated from each other in order to avoid problems due to crowding/blending. The code carries out photometry on this frame and recovers positions and instrumental magnitudes of stars. The positions of the stars obtained after photometry are converted to parsec (from pixels) and then matched with the positions of the stars in the original list of the stars in order to compare the obtained instrumental magnitude for those stars with their actual absolute magnitudes. Using the differences between the instrumental magnitude and the actual magnitude of the stars, a linear fit can be done that provides us with a correction term to convert our instrumental magnitude to absolute or apparent magnitudes. This procedure is also useful in getting a good PSF model for a very dense cluster model. As the stars are well separated in this image, the PSF model obtained during the photometry of this frame has a lower error value and this PSF model can then be used to fit all the frames that collectively image the simulated cluster models.

[3] This program to find the FWHM was developed by Wojciech Pych and has been used in determining FWHM of actual observational images of star clusters for the Cluster AgeS Experiment (CASE; Pych et al. 2001; Kaluzny et al. 2005; Rozyczka et al. 2017, and references therein).

Table 1. Initial properties, mass, and binary fraction at 12 Gyr for the model for which the CMD is shown in Fig. 3. The initial mass function (IMF) for this model is a three segment Kroupa IMF (Kroupa, Tout & Gilmore 1993). Black holes and neutron star natal kicks were drawn from a Maxwellian distribution with $\sigma = 265\,\mathrm{km\,s^{-1}}$ (Hobbs et al. 2005).

Model	Initial number of objects	Initial mass ($M_\odot$)	Initial binary fraction (per cent)	Galactocentric radius (kpc)	12 Gyr mass ($M_\odot$)	12 Gyr binary fraction (per cent)
Tidally filling	1.12×10^6	8.07×10^5	95	8.0	3.86×10^5	28.8

This gets rid of the errors associated with differences in PSF models obtained when numerous frames are required to image the entire cluster and the FWHM and PSF are computed individually for each frame.

3 COCOA IN ACTION: APPLICATIONS AND RESULTS

3.1 Mock observations and photometry of a cluster model

In this section, we present results from COCOA for a tidally filling cluster model at the time 12 Gyr. The model was evolved using the MOCCA code. The snapshot at 12 Gyr was used as an input file in COCOA and the projected snapshot with positions, velocities, and magnitudes with respect to the centre of the cluster was created. Then COCOA and the SIM2OBS tool in it were used to automatically image the entire cluster. Photometry using our automated pipeline was carried out and CMD were created for the star cluster. In this section, the details of how mock observations are generated and used to obtain photometric results are provided.

From the results of MOCCA survey, a star cluster model with initially $N = 1.12 \times 10^6$ objects, low concentration, high initial binary fraction of 95 per cent, and a metallicity of $Z = 0.001$ was selected. High initial binary fraction with properties given by the Kroupa initial binary parameter (IBP; Kroupa 1995b) for GC models can reproduce observed present-day binary fraction and the observed anticorrelation between the binary fraction and cluster mass (Leigh et al. 2015; Belloni et al. 2017). Initial conditions for this model were not explicitly selected to reproduce a particular Galactic GC, the main idea was to look at the evolution of stellar/binary populations in a sparse cluster in which dynamical interactions were not too frequent. The IBP distribution for this model used the Kroupa IBP for the semimajor axis and eccentricity distributions but had an almost flat distribution for the mass ratio. The model was simulated in an external potential assuming a point-mass potential for the Milky Way at a Galactocentric radius of 8 kpc. At 12 Gyr, the model has $N = 9.67 \times 10^5$ objects with a 28.8 per cent binary fraction and is extended up to 60 pc. The high binary fraction at 12 Gyr is dominated by binaries in which at least one component is less massive than $0.3\,M_\odot$. Initial and 12 Gyr properties for the model are provided in Table 1. We used the projected x, y positions and the magnitudes for each star in V band to generate synthetic observations of the cluster. The cluster was observed at a distance of 5 kpc. In the next subsection (Section 3.1.1), the details of the parameters that were used to simulate the observations for this cluster are provided.

3.1.1 *Telescope parameters*

The parameters of instrument with which the mock observations were created were extremely idealized with very low seeing values, high spatial resolution, and a Gaussian PSF. Instrument specifications were chosen to match those of a typical 8 m class telescope. The pixel scale that we specified in SIM2OBS was 0.10 arcsec pixel^{-1} and the size of the CCD was 4096×4096 pixels. In order to image the entire cluster up to a radius of 60 pc from the centre of the cluster, the COCOA code automatically generated offsets (using the size of the CCD, pixel scale, and distance to the cluster) and parameter files for SIM2OBS. The code then created 169 FITS files in order to effectively image the cluster up to a radius of 60 pc ($\sim$2475 arcsec). The spatial resolution of this simulated instrument is fairly high and the field of view (FOV) of each frame is 409.6 arcsec. The saturation level for the synthetic observation was kept artificially large in order to avoid too many saturated pixels and have as ideal observations as possible for high exposure times. The important parameter values provided to SIM2OBS in order to image the cluster in the V-band filter are given below (see Appendix A3 for more details about these parameters).

(i) DISTANCE = 5000 [pc].
(ii) PIXSCALE = 0.10 [arcsec pixel^{-1}].
(iii) GAIN = 0.9 [photons ADU^{-1}].
(iv) SATLEVEL = 990000.0 [counts].
(v) EXPOSURE = 35 [multiple of direct counts].
(vi) SEEING = 0.3 [arcsec].
(vii) BACKGROUND = 1.
(viii) PSF = G [Gaussian profile was used for the PSF model].
(ix) NOISE = 1 [Poisson noise was enabled].

Similar values were used for generating the B-band images. However, a higher exposure value of 65 (multiple of direct counts was used) as stars are dimmer in the B band and require longer exposure to be imaged. Also when using the DAOPHOT routine to pick the stars for making the PSF model, a brighter limiting magnitude for B band was used.

3.1.2 *V- and B-band photometry*

Once the FITS images were produced for both the V- and B-band filters, the automated PSF photometry pipeline in COCOA was used to create a catalogue of all stars that contains their magnitudes and (error in magnitudes) using the procedures explained in Section 2.3 and Appendices A4 and A5. The values in the option files for DAOPHOT and ALLSTAR for these models can be found in Table 2.

From photometry of our simulated FITS images, we were able to recover 201 005 stars in the V band and 147 966 stars in the B band. From the simulation snapshot, we know that the actual number of stars were 966 998. The number of recovered stars depends significantly on the exposure value that is used. More stars can be recovered with higher exposure values. Fig. 2 shows the magnitude versus error plots in V- and B-band filters obtained from the PSF photometry of the mock observations.

3.1.3 *Colour–magnitude diagram*

Once the catalogues from the V- and B-band photometry were obtained, we joined the data for stars that had the same positions

Table 2. Values of parameters in DAOPHOT/ALLSTAR option files for the test model. The values in these option files were generated using the prescriptions explained in Section A5. The FWHM of our simulated images was 3.0.

Parameters in daophot.opt	Parameters in photo.opt	Parameters in allstar.opt
FWHM = 3.02	A1 = 4.53	fit = 3.624
FIT = 3.624	IS = 12.08	isky = 6.04
PSF = 8.154	OS = 15.1	osky = 15.1
READ = 0.1		watch = 0
GAIN = 0.9		redet = 1
TH = 5.0		
AN = 1		
LOWBAD = 15		
HIBAD = 990000.0		
WATCH = −2.0		
VAR = 0		

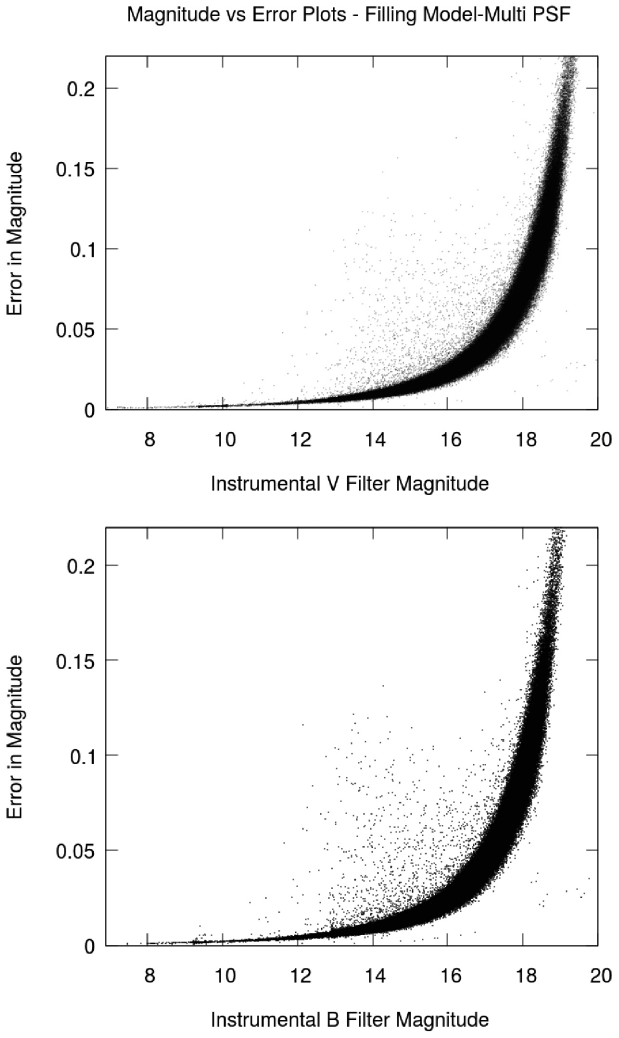

Figure 2. Error versus magnitude plot for *V*- and *B*-band photometry. The *x*-axis represents the instrumental magnitude and the *y*-axis represents the error in magnitude.

(within 0.01 arcsec) in both the catalogues. The joined data allowed us to create a CMD of our model (Fig. 3). We compared this observed CMD with the one obtained directly from the simulation data and we can see that the observed CMD well recovers the simulation CMD till up to 3–4 mag below the main sequence (MS) turn-off.

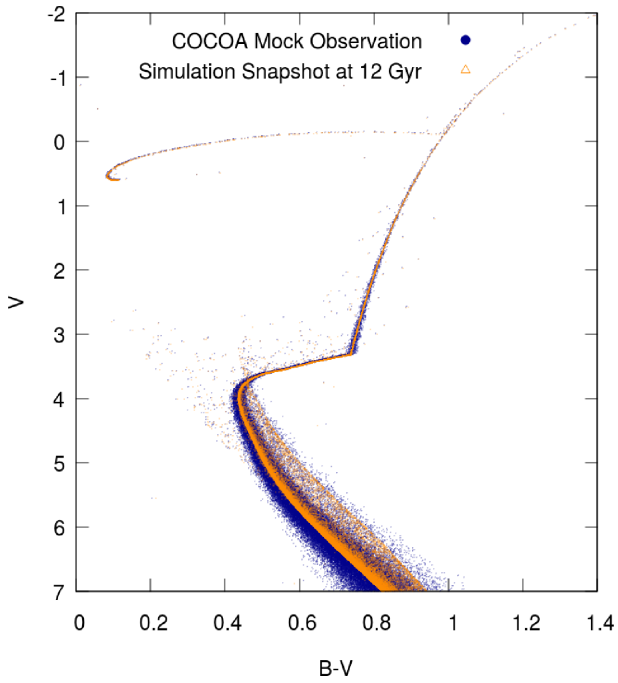

Figure 3. Colour–magnitude diagram (CMD) for the cluster model at 12 Gyr. The magnitudes from observations have been scaled to absolute magnitudes by comparing it with the CMD taken from the simulation snapshot. The $B - V$ colour is on the *x*-axis and the *V*-band magnitude is on the *y*-axis. The blue points indicate observed stars from our PSF photometry pipeline. The orange points are from the CMD directly taken from the snapshot of our simulation results.

3.2 Testing photometry pipeline with actual observations

In order to ensure that the automated photometry module was working accurately and to check the validity of the parameters used in the COCOA code for working out synthetic images, we used the photometry module to carry out photometry on actual *B*- and *V*-band observations of a real star cluster. Photometry was also performed on the same observations independently by observers using their own parameters.

The observations were made using the ESO Wide Field Imager (WFI) mounted on the 2.2 m telescope in La Silla, Chile. The particular observations in *V* and *B* bands imaged part of the Galactic GC NGC 2808. Photometry was done on both the *V*- and *B*-band images using the automated photometry module in COCOA. A CMD was created and compared with the CMD obtained independently by observers (Dalessandro et al. 2011). Fig. 4 shows the two CMDs for the same observation, the orange points are from the photometry results from COCOA, and the blue points are from the photometry done by the observers. We also compared the photometric errors from the two different procedures and found that COCOA can do reasonably well photometry with its automated pipeline. There is a scatter in the CMD obtained from COCOA results that can be seen with the orange points in Fig. 4. This scatter is connected with the incorrect matching of stars from the catalogues in two different filters. Nearly 20 per cent of the stars from both catalogues were not properly matched, nearly all these stars were from the dense part of the observed image (left-hand panel in Fig. 5). The match in

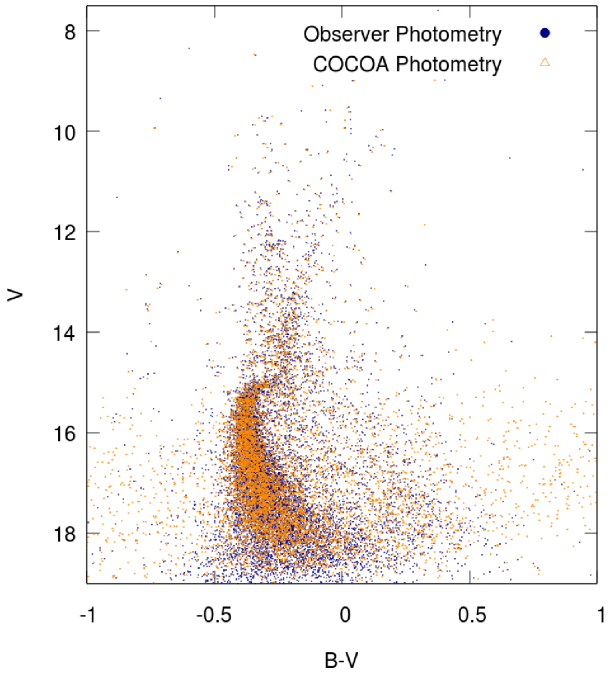

Figure 4. Colour–magnitude diagram for an actual observation of the Galactic GC NGC 2808. The magnitudes are instrumental and the orange points show the results from the module in COCOA, while the blue points are the photometry results done by observers.

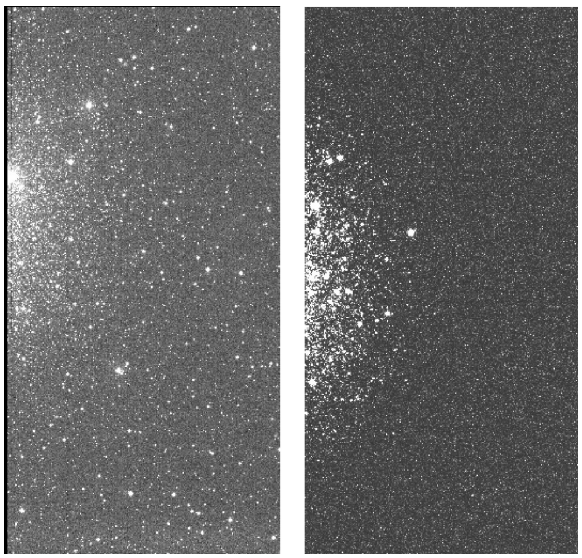

Figure 5. Left: observational image of the Galactic GC NGC 2808. The observation was carried out in the V/89 filter of the ESO WFI that is mounted on the 2.2 m telescope in La Silla, Chile (Dalessandro et al. 2011). Right: synthetic observational image created using COCOA of a GC model with observational properties comparable to NGC 2808. The parameters for creating this mock observational image were set to replicate the properties of the WFI instrument mounted on a 2.2 m telescope.

the CMD can be made more accurate by improving the criteria to match stars from the photometry results from two different filters in COCOA. The matching routines will be further improved to take into account the magnitudes of stars in the next version of the COCOA code.

3.3 Mock observations of a cluster model similar to NGC 2808

In the previous subsection, we showed that the photometry pipeline in COCOA can be used to reduce observational data. In this subsection, we compare the observation of NGC 2808 with simulated mock observations of a star cluster model created using COCOA with the same age and metallicity as NGC 2808. We use the snapshot at 11 Gyr from a GC model that was simulated using the MOCCA code and has total luminosity and tidal radii that are comparable to NGC 2808 in order to simulate mock observations with COCOA. The model was initially selected from a data base of 1948 simulated cluster models as part of the MOCCA Survey Database I (Askar et al. 2017b) project. The model was then resimulated with a larger number of stars ($N = 2.2 \times 10^6$) and the metallicity for the simulated cluster model was set to [Fe/H] $= -1.18$ ($Z = 0.00132$) to match the observed metallicity of NGC 2808 (Carretta et al. 2009). For such a comparison, the metallicity value provided to the SSE and BSE codes for binary stellar evolution in MOCCA is important, as the fundamental properties of the star-like mass, radius, and luminosity that will determine their magnitudes depend significantly on this value and the age of the cluster.

The 11 Gyr snapshot was taken from the output of the MOCCA simulation. The projection module in COCOA was used to project this snapshot. The properties of the stars (mass, luminosity, and effective temperature) from the simulation snapshot were provided to the GALEVNB code (explained in Section 3.4) to obtain absolute magnitudes in ESO WFI specific V/89 and B/99 filters for each star. Using these magnitudes and the projected positions of the stars, COCOA was used to create mock observations that were simulated with parameters set to replicate the ESO WFI mounted on a 2.2 m telescope. The FOV was offset to approximately replicate the observed pointing and the distance was set to 9.6 kpc to match the observed distance to NGC 2808 (Harris 1996, updated 2010). Observations were simulated in *B* and *V* band with the same exposure times and seeing value as the actual observations. Fig. 5 shows the actual observational image (for the *V* filter) that was used for this comparison and the mock image of the cluster model at 11 Gyr created with MOCCA. The parameters provided to the SIM2OBS code to obtain the V/89 filter mock observation in the observation/imaging module of COCOA are listed below.

(i) DISTANCE = 9600 [pc].
(ii) NAXIS1 = 2048 [image width in pixels].
(iii) NAXIS2 = 4096 [image length in pixels].
(iv) PIXSCALE = 0.238 [arcsec pixel^{-1}].
(v) GAIN = 2.0 [photons ADU^{-1}].
(vi) SATLEVEL = 100000.0 [counts].
(vii) EXPOSURE = 120.0 [multiple direct counts].
(viii) SEEING = 0.78 [arcsec].
(ix) BACKGROUND = 1.
(x) PSF = M [Moffat distribution for the PSF model].
(xi) M_BETA = 1.6 [beta parameter for Moffat distribution].
(xii) NOISE = 1 [Poisson noise was enabled].

For the B/99 band observations, most of the above parameters were the same but the exposure value was changed to 180.0 to match the exposure time of the real B/99 filter observation. After the mock observational images were generated in the two filters, we used the photometry module in COCOA to carry out PSF photometry on the images and obtain catalogues of stars. The recovered instrumental magnitudes were first converted to absolute magnitudes by comparing the instrumental magnitudes with the absolute magnitudes provided by GALEVNB. The absolute magnitudes were then

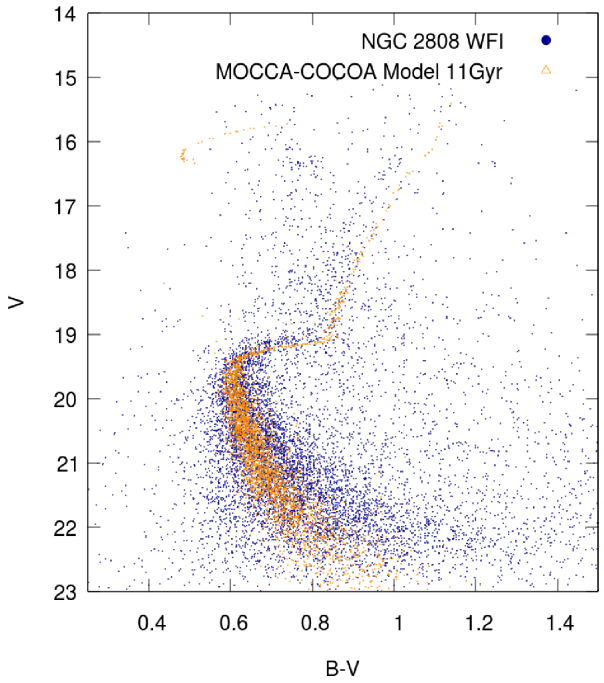

Figure 6. Colour–magnitude diagram showing the comparison between the results of the actual observation of NGC 2808 (blue points) with that of a mock observation of a simulated cluster model at 11 Gyr with the same metallicity as NGC 2808 (orange points).

converted to apparent ones using the distance modulus (15.23) and reddening value ($E(B-V)=0.17$) used by the observers (Dalessandro et al. 2011). We then matched the stars from the two different filters based on the recovered position of the stars to obtain a CMD that we could then compare with the observational CMD.

Fig. 6 shows the result of this comparison between the observation of NGC 2808 and of a cluster model with the same age and metallicity as NGC 2808. From the comparison it can be seen that the photometry from the mock observation overlaps quite well with the actual observation. In this particular case, the actual observation is plagued by background stars, however, the positions of the MS turn-off match quite well in both the CMDs. There are some minor differences with regards to the extension of the subgiant branch in colour and the position of the base of the giant branch. We carefully investigated these differences and attribute them to the properties of the stars provided by the SSE and BSE codes that use analytic fitting formulae based on the stellar models computed by Pols et al. (1998). For low metallicity values typical of GCs, the subgiant branch for a set of stars evolved with SSE is slightly extended in the redder colour and subsequently the base of the red giant branch also shifts to a redder colour compared to their positions in actual observations and also in CMD isochrones based on newer stellar evolution track models such as PARSEC isochrones (Bressan et al. 2012). These differences in the stellar properties can have an influence on the global and specific properties of star clusters (such as luminosity function, surface brightness profile) that are obtained from simulated models. Moreover, these differences should be taken into account when making detailed comparisons of simulated models and observations for the purpose of constraining initial conditions for particular GCs. The population synthesis code SEVN developed by Spera, Mapelli & Bressan (2015) reads and interpolates a grid of PARSEC stellar evolution isochrones to determine stellar properties for massive stars. Such an approach that extends to lower masses can be useful for stellar dynamic codes to get stellar properties based on the latest evolution models. This also demonstrates that using COCOA to compare results of simulated models with actual observations can be helpful in determining and understanding the shortcomings of stellar evolution models and codes.

3.4 COCOA and GALEVNB

COCOA can be used together with the publicly available GALEVNB code[4] (Kotulla et al. 2009; Pang et al. 2016) to obtain magnitudes in different photometric filters for stars in simulation snapshots. The GALEVNB code can provide magnitudes for stars in snapshots of numerical simulations in a variety of different filters for various wavelengths. Using stellar properties such as mass, temperature, luminosity, and metallicity, GALEVNB can generate a spectra that covers far-ultraviolet (UV) to far-infrared (IR) wavelengths with a resolution of 20 Å (Pang et al. 2016). To obtain magnitudes in different filters, GALEVNB convolves the spectra with the filter response functions and applies selected zero-points corrections.

The projected snapshot generated by COCOA can be used with GALEVNB in order to obtain absolute magnitudes for stars in MOCCA snapshots in different filters, this allows us to simulate multiband observations of simulated cluster models. GALEVNB can also be used by *N*-body modellers (working with NBODY6/NBODY6++GPU to obtain absolute magnitudes for their simulation snapshots in multiple filters. Having absolute magnitudes and projected positions of stars, the imaging and photometry modules in COCOA can be used to create mock observations and photometry on these mock images from *N*-body snapshots. Having magnitudes in instrumental filters is particularly useful as it can allow for meaningful comparison with observational studies and surveys that use particular CMDs to determine binary fractions and multiple population sequences in GCs.

3.5 Multiband photometry with COCOA and IBP constraints

In this section, we show results from multiband photometry of sparse and moderately dense cluster models that were evolved using MOCCA. The snapshot of the clusters after 12 Gyr of evolution was used to simulate FITS images in at least five photometric bands (*U*, *B*, *V*, *R*, *I*). The automated PSF photometry pipeline in COCOA was then used to obtain catalogues of stars from these simulated models.

All cluster models that were used had initially large N (1×10^6 stars) and metallicity $Z=0.001$. Models beginning with the prefix K have a high initial binary fraction of 95 per cent and initial binary properties were based on Kroupa's IBPs (Kroupa 1995b). The cluster models beginning with the suffix S have a primordial binary fraction of 5 per cent (S1) and 10 per cent (S2). Binary parameters for S models had 'standard' distributions [uniform mass ratio (q) distribution, thermal eccentricity distribution, and log-normal or uniform in log semimajor axis distribution]. Table 3 summarizes the initial and 12 Gyr properties of these cluster models that have also been described in Belloni et al. (2016).

Fig. 7 shows the simulated *I*-band image for the moderately dense K2 cluster projected at a distance of 5 kpc and Fig. 8 shows CMDs for this model.

Multiband simulations were also done for sparse models with high binary fraction and Kroupa distributions for the IBP (K1). As

[4] GALEVNB can be obtained from http://silkroad.bao.ac.cn/repos/galevnb/.

Table 3. The initial properties of the models and their 12 Gyr binary fraction.

Model	Mass ($M_\odot$)	Initial binary fraction (per cent)	Central density ($M_\odot$ pc^{-3})	r_t (pc)	r_h (pc)	12 Gyr binary fraction (per cent)
K1	8.07×10^5	95	1.9×10^2	115	16.9	28.3
K2	8.07×10^5	95	7.8×10^4	115	2.3	8.9
S1	5.92×10^5	05	2.8×10^3	97	7.5	4.9
S2	9.15×10^5	10	1.3×10^5	125	2.1	4.8

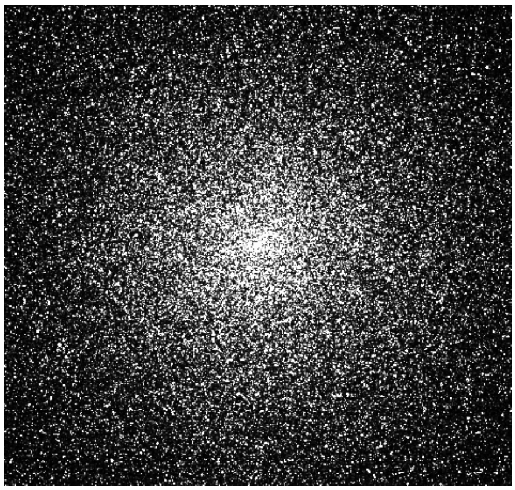

Figure 7. *I*-band image of the innermost frame imaging the core of the cluster model K2. 2048 × 2048 pixels. The cluster was imaged with a high spatial resolution telescope with pixel scale of 0.08 arcsec pixel^{-1}. The projected distance to cluster model was 5 kpc.

this model was sparse and the initial binary fraction was very high, a lot of binaries survived up to 12 Gyr. The second sequence due to the presence of binaries with $q \approx 1$ is very prominent in all CMDs of this model (Fig. 9). Binaries with nearly equal mass ratios can result in a second sequence that has an apparent turn off point about 3/4 of a magnitude above the turn off point for single MS stars. The broadening of the MS by binaries of different mass ratios has been thoroughly addressed by Hurley & Tout (1998). As stated above, the second sequence due to the binaries with nearly equal mass components is prominent for models with high primordial binary fractions and initial parameter distributions given by Kroupa (1995a) and Marks & Kroupa (2012). This is because this primordial binary distribution of the mass ratio has a disproportionately large number of low-mass binaries with q being very close to 1 due to mass feeding during pre-MS eigenevolution proposed by Kroupa (1995a). This issue with the second sequence in the IBPs given by Kroupa (1995a) has been addressed in Belloni et al. (2017) and a revised prescription for the mass feeding during pre-MS evolution has also been proposed.

This second sequence is more prominent when the model is sparse and less prominent in denser models where even primordial binaries with low-mass components are more likely to be disrupted.

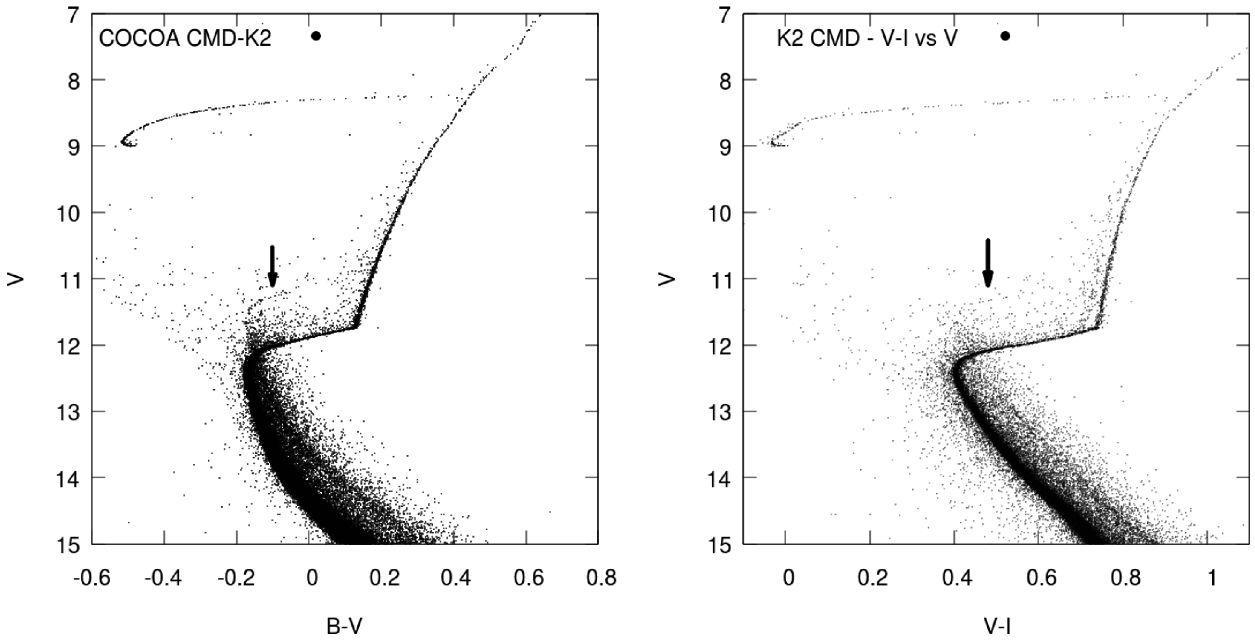

Figure 8. Left-hand panel: shows the $B - V$ versus V CMD for the K2 model using instrumental B- and V-band magnitudes. There are indications of the presence of a second MS due to many binaries with q (mass ratio) close to unity. The black arrows show these binaries that are visible above the MS turn-off. Right-hand panel: shows the $V - I$ versus V CMD for the same model that is significantly denser than the K1 model. Because of this higher density and overall lower binary fraction, the photometric errors are larger and it becomes difficult to distinguish the second sequence. Also with higher density, a significant number of such binaries are disrupted. However, there are still binaries with nearly equal mass components above the MS turn-off that are indicated with the black arrow.

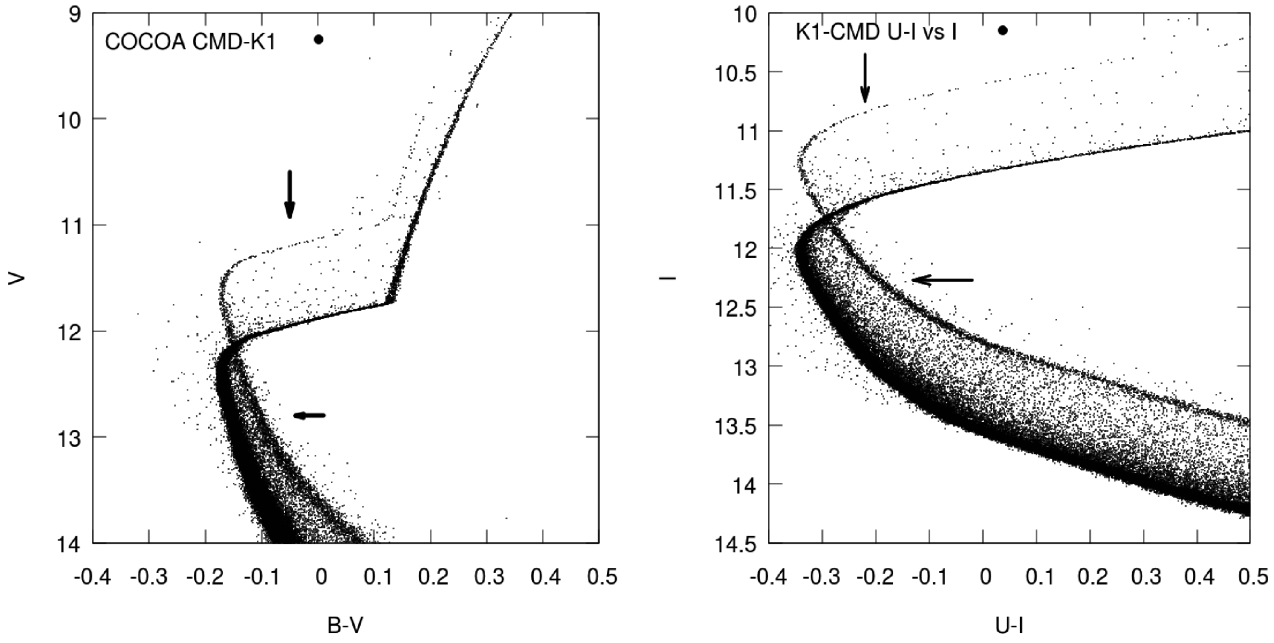

Figure 9. Left-hand panel: $B - V$ versus V CMD for the K1 model produced using instrumental B- and V-band magnitudes. There is a prominent second MS due to many binaries with q close to unity that is indicated with the black arrows. Right-hand panel: $U - I$ versus I CMD for the K1 model. The figure zooms in on part of the CMD close to the MS turn-off. The presence of a large number of binaries (with q close to 1) causes a second sequence that has its own turn-off 3/4 of a magnitude above the turn-off for single stars.

Moreover, the photometric errors are larger in denser fields and the second sequence becomes more difficult to see in the CMD due to these errors. By determining the inferred binary mass ratios from the broadening of the MS, we can determine whether the IBPs we use in the simulated cluster models can recover the observed mass ratio distribution for binaries observed in present day GCs. If the simulated cluster models with particular initial binary fraction and parameter distribution show features that are not corroborated by observational results, then these parameters may require further corrections. This is an important application of COCOA that can be helpful in improving theoretical models and initial conditions for star cluster simulations. Apart from second MSs, other differences between mock observations of star cluster models and actual observations (e.g. the number and observational properties of blue stragglers, number of evolved giants) may also be used to improve cluster initial conditions.

The second sequence is also very prominent in the $U - I$ versus I CMD for the K1 model (this can be seen in the right-hand panel of Fig. 9). A combination of UV and visual band filters can be particularly useful for identifying elemental abundances and variations for stellar population in clusters (Bellini et al. 2010).

For models with standard binary parameter distribution (S-models), no multiple sequences are seen for CMDs in different filters. $B - V$ versus V CMDs are shown for models S1 and S2 in Fig. 10.

For the sparse model with standard IBP distributions, the absence of the second sequence due to $q = 1$ binaries can be clearly seen in the left-hand panel of Fig. 10.

3.6 Imaging distant clusters

In this section, we briefly present results from the K1 cluster model for which mock observations were produced with COCOA at a distance of 20 kpc. Photometry was carried out on these mock observations using COCOA and the CMDs for this model still shows a prominent second sequence due to binaries with large mass ratio values. Despite photometric errors being higher when the cluster was observed at a large distance, the second sequence is fairly prominent because the cluster model is quite sparse. Fig. 11 shows CMDs $B - V$ versus I and $V - I$ versus V for the K1 model observed at a distance of 20 kpc.

To check the influence of cluster density and distance on recovering photometric features. We also created mock observations for the more dense K2 model by observing the cluster at a distance of 15 kpc. Photometric errors were higher for this cluster model as the field is more crowded. Because of the density of the cluster and the photometric noise, it can be difficult to determine whether there could be multiple sequences being caused by high mass ratio binaries. The two panels in Fig. 12 show the $B - V$ versus V and $V - I$ versus V CMDs for the K2 model projected at a distance of 15 kpc. These results show that for dense models, photometric features like a second sequence due to binaries with nearly equal mass components can only be resolved for nearby clusters. For sparse models, such a feature may even be resolved for a distant cluster. This is just to show one of the many possibilities of how the distance to a cluster and its density may influence the observational results. This can be particularly useful for determining whether certain population of stars, binary fractions, and other properties can be properly recovered for very distant cluster and how those results would change if that cluster was observed at a closer distance.

4 FUTURE DEVELOPMENT AND IMPROVEMENTS

There are a variety of uses and applications for the COCOA code and some of these were shown in the previous section. However, there are numerous features that will be further developed to improve comparison between simulations and observations of star clusters.

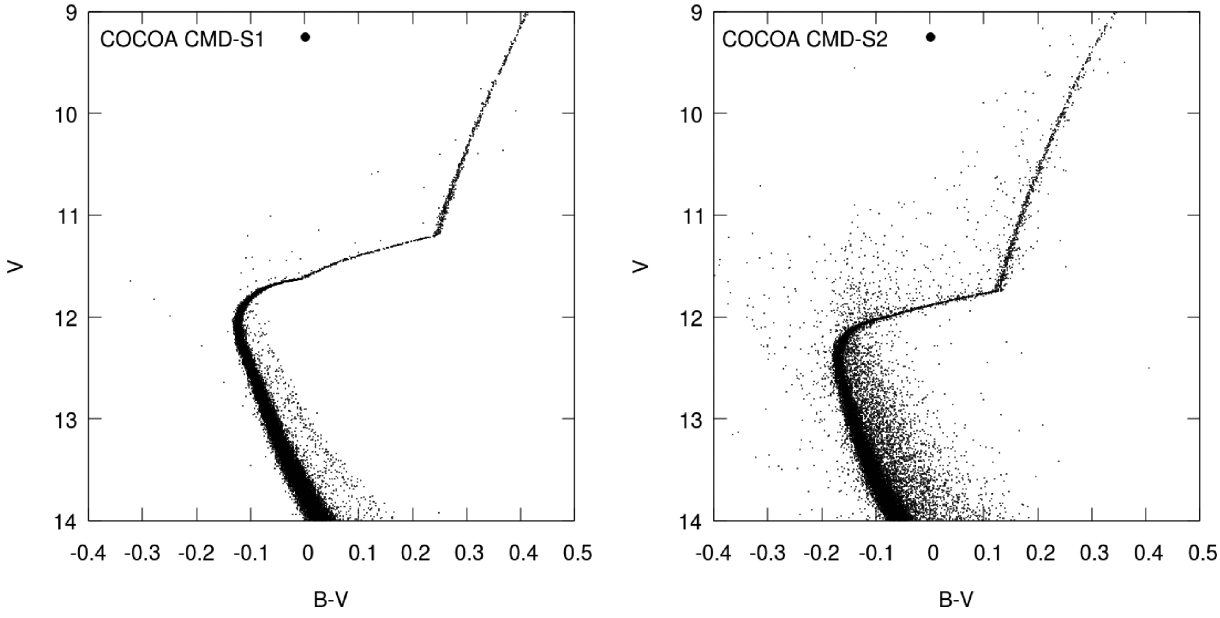

Figure 10. Left-hand panel: $B - V$ versus V CMD for the S1 model. Even though this model is initially sparse, no second sequence due to equal mass ratio binaries is seen. CMDs produced using other filters also showed an absence of the second sequence. Right-hand panel: $B - V$ versus V CMD for the S2 model. Similar to model S1, there is no presence of a prominent second sequence due to equal mass binaries that was seen in models K1 and K2.

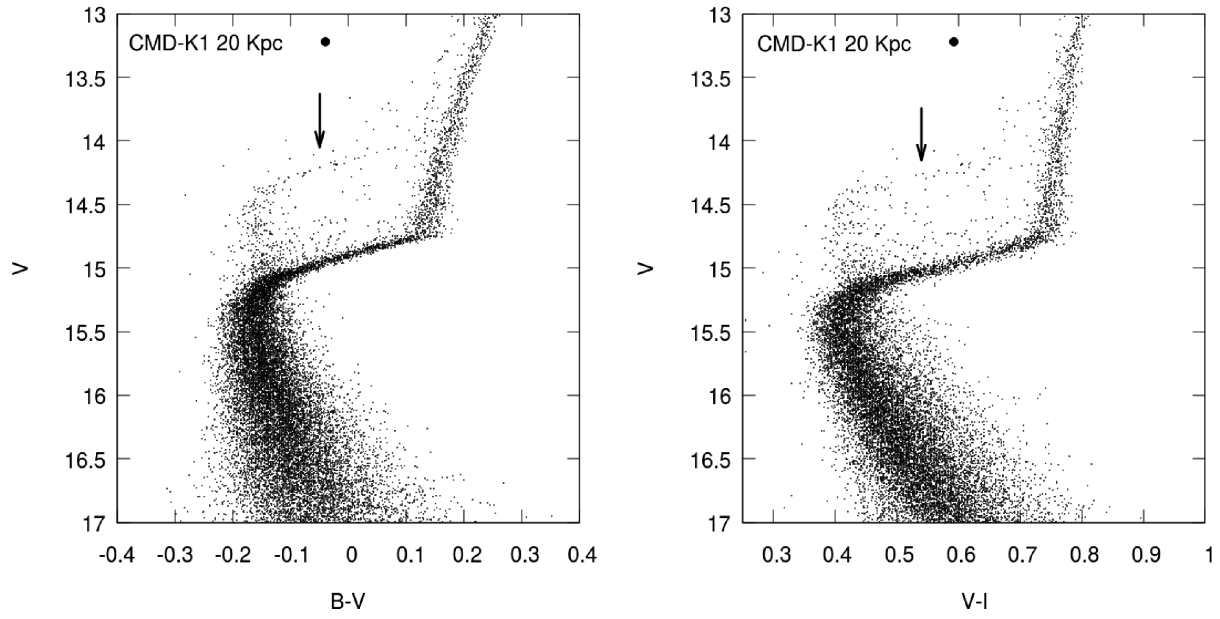

Figure 11. Left-hand panel: $B - V$ versus V CMD for the K1 model. The cluster was projected at a distance of 20 kpc and the spatial resolution of the instrument was 0.08 arcsec pixel^{-1}. The second sequence due to binaries with nearly equal mass components is visible above the MS turn-off and is indicated with the black arrow. Right-hand panel: $V - I$ versus V CMD for the K1 model using instrumental V- and I-band magnitudes. The second sequence above the MS turn-off is indicated with the black arrow.

The code can already simulate the movement of stars in binary systems on short time-scales and generate a sequence of projected snapshots for which observations can be simulated. Accurate treatment of changes in magnitudes for such systems during eclipses will be very useful in estimating the number of eclipsing binaries that maybe observed in particular star clusters. Moreover, this accurate treatment of changes in magnitude due to the movement of the stars in binaries can also be helpful in identifying and observing exotic objects like cataclysmic variables.

For a sequence of projected snapshots generated on short time-scales, it will also be possible to use the velocities of all single and binary stars to change their positions according to their proper motions on the plane of the sky. With photometric analysis of these sequence of snapshots, proper motions for the stars in the cluster can be determined. This can be particularly useful for determining global properties of GCs. This approach with tracking the movement of the stars in the projected cluster snapshots will also open up the possibility of simulating and detecting occultations of stars by other cluster members.

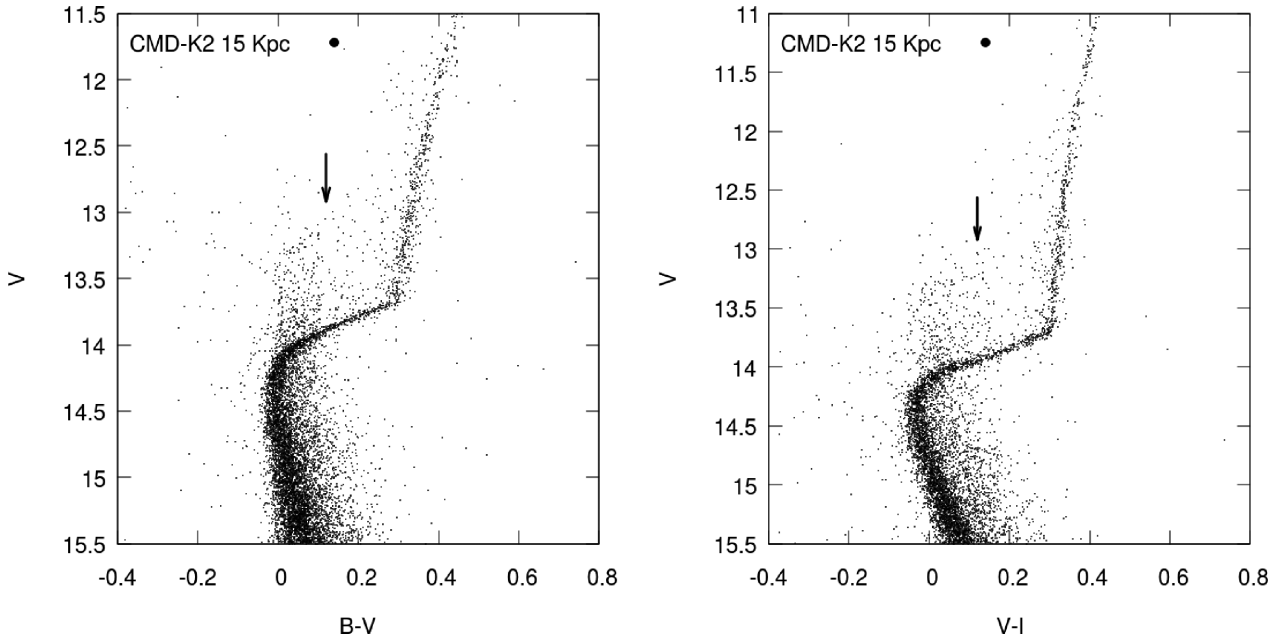

Figure 12. Left-hand panel: $B - V$ versus V CMD for the K2 model using instrumental B- and V-band magnitudes. The cluster was projected at a distance of 15 kpc and the spatial resolution of the instrument was 0.08 arcsec pixel^{-1}. The black arrow indicates the second sequence due to $q \approx 1$ binaries above the MS turn-off. Right-hand panel: $V - I$ versus V CMD for the K1 model using instrumental V- and I-band magnitudes. The second sequence above the MS turn-off is indicated with the black arrow.

For future development it will be very important to introduce more automated procedures to as accurately as possible apply observational techniques to our simulated cluster models. For instance, proper aperture correction procedures can be implemented when photometric catalogues from multiple images are being combined. Particularly interesting would be to determine binary fractions in the same way as it is done by observers (Milone et al. 2012). This will be helpful in determining the accuracy of the observational technique and influences of photometric errors as values obtained from it can be directly compared with simulations. Automated procedure to use observational methods to determine cluster centres from photometric observations can also be applied to our simulated images to determine the validity in determining the cluster centre. This is can be particularly important for obtaining accurate velocity dispersion and other profiles.

It was mentioned in Section 3.2 that we need to improve the matching criterion for stars from two different filter catalogues. Comparing positions might not be sufficient especially when the field is extremely dense, therefore, a more sophisticated routine that can also compare magnitudes will be required to match the stars. It will be useful to have this feature in COCOA to match the stars and produce CMDs automatically for the users.

There are also plans for integrating the SISCO (Bianchini et al. 2015) code that can create mock kinematic observations of simulated star cluster models with COCOA. SISCO has already been modified to use the projected snapshot from COCOA to produce simulated IFU observations of star cluster models. Both COCOA and SISCO were recently used to simulate photometric and kinematic observations of a dark star cluster model in Askar et al. (2017a). Currently, COCOA can generate surface brightness and velocity dispersion profiles (produced from the projected snapshot) that can be fit to simple dynamical models like King (1966). In the future, we plan on introducing options to fit profiles to different models such as Jean's model or more sophisticated multimass models such as those developed by Gieles & Zocchi (2015)[5], de Vita, Bertin & Zocchi (2016), and Peuten et al. (2017). With integration of SISCO, it is possible to obtain mock observational kinematic profiles that will further improve comparison with observations and can help determine the accuracy of these models in recovering cluster parameters.

COCOA will also be integrated within the Virtual Observatory Star Cluster Analysis (VOSCA) service[6] that is part of the AstroGrid-PL tools to allow users to create synthetic observations of particular models from a large data base of MOCCA and N-body models. AstroGrid-PL[7] is a Polish project that has been developed to exploit latest computational technology to provide domain-specific grid services and tools that can be used for research in astronomy and astrophysics.

5 CONCLUSIONS

Creating mock observations of star cluster models and comparing them with observations has numerous advantages. Not only is this useful in seeing how cluster models would appear as observations but also has significance in checking the validity of the observer's data reduction processes. It is also particularly useful for comparing simulated models with real star clusters, which allows one to constrain initial conditions for star cluster models and rule out values that do not match with observations. The purpose of COCOA is to provide theoreticians with an easy to use tool that can help them present their results in a way that is most meaningful for observers. This paper demonstrated the various applications of COCOA and the insights that one can gain from observationally viewing simulated star clusters. Problems with binary parameter distributions,

[5] LIMEPY code developed by Gieles & Zocchi (2015) can be obtained from https://github.com/mgieles/limepy.
[6] https://astrogrid-pl.org/vo/user_doc/using_vosca
[7] https://astrogrid-pl.org

initial conditions, and cluster parameters may not be evident from numerical data of evolved clusters. Simply simulating ideal mock observations sheds new light on that data and with theoreticians producing more realistic simulations, the comparison with observations becomes even more necessary. COCOA can also be significantly important for observers that would like to make use of interesting theoretical results and check whether observations can verify those results. For instance, a number of numerical simulations predict expected number of various exotic objects like cataclysmic variables (Belloni et al. 2016), blue stragglers (Hypki & Giersz 2017), and other interesting objects. With COCOA, it is possible to check whether such predicted numbers of exotic objects and their spatial distribution can be observationally recovered. Previous works (Chatterjee et al. 2013; Sills et al. 2013; Pasquato & Chung 2016; de Vita et al. 2017; Geller et al. 2017; Sollima et al. 2017) that have carried out such comparisons between observation and theory have been able to provide important constraints and results. Furthermore, recent MOCCA simulations by Giersz et al. (2015) have predicted the formation of a intermediate-mass black hole (IMBH) in GCs, particularly interesting are models in which the IMBH dominates the present-day mass of the cluster. COCOA has been used to check whether such dark star clusters are observed in the Galaxy. In Askar et al. (2017a) a comparison for the observed photometric properties of a simulation model was carried out using COCOA that together with mock kinematic observations from SISCO (Bianchini et al. 2015) helped to identify NGC 6535 as a possible candidate for harbouring an IMBH.

Automating observational data reduction and providing end results to users will be particularly useful for theoreticians who are not familiar with observational procedures. With COCOA, all the steps including PSF photometry have been automated. Users with observational experience have the option to use modules that they require and generate mock observational images of the cluster and then perform photometry on them using their preferred tools and scripts. Another motivation for automating the procedures in COCOA is to be able to simulate and analyse mock observations of about 2000 models that were simulated as part of the MOCCA Survey Database project (Askar et al. 2017b). More surveys of star clusters models will be carried out with the Monte Carlo code MOCCA in the near future. Analysing models with different initial conditions using COCOA can help in improving simulations to reproduce observed cluster models. COCOA will be made publicly available and it can be further developed by the community to incorporate more realistic synthetic observations.

ACKNOWLEDGEMENTS

We are extremely grateful to the reviewer for providing a very thorough report that was very informative and gave suggestions that have been very helpful in improving the presentation, quality, and content of the paper. We would like to thank Ralf Kotulla for making the GALEV code accessible for generating magnitudes in different filters. We would also like to thank Douglas Heggie for providing the code that fits the King (1966) model to surface brightness and velocity dispersion profiles. We are very grateful to the following people for providing fruitful discussions, feedback, resolving technical difficulties, and explaining observational procedures that helped in developing COCOA: Weronika Narloch, Krystian Ilkiewicz, Arkadiusz Olech, Grzegorz Pietrzyński, Barbara Lanzoni, Ammar Askar, Arkadiusz Hypki, Diogo Belloni, and Paolo Bianchini. AA and MG were supported by the National Science Centre (NCN) through the grant DEC-2012/07/B/ST9/04412. AA would also like to acknowledge support by NCN through the grant UMO-2015/17/N/ST9/02573 and support from Nicolaus Copernicus Astronomical Center's grant for young researchers.

REFERENCES

Askar A., Giersz M., Pych W., Olech A., Hypki A., 2016, in Meiron Y., Li S., Liu F., Spurzem R., eds, Proc. IAU Symp. Vol. 312, Star Clusters and Black Holes in Galaxies across Cosmic Time. Cambridge Univ. Press, Cambridge, p. 262
Askar A., Bianchini P., de Vita R., Giersz M., Hypki A., Kamann S., 2017a, MNRAS, 464, 3090
Askar A., Szkudlarek M., Gondek-Rosińska D., Giersz M., Bulik T., 2017b, MNRAS, 464, L36
Bellini A., Bedin L. R., Piotto G., Milone A. P., Marino A. F., Villanova S., 2010, AJ, 140, 631
Belloni D., Giersz M., Askar A., Leigh N., Hypki A., 2016, MNRAS, 462, 2950
Belloni D., Askar A., Giersz M., Kroupa P., Rocha-Pinto H. J., 2017, MNRAS, 471, 2812
Bianchini P., Norris M. A., van de Ven G., Schinnerer E., 2015, MNRAS, 453, 365
Bressan A., Marigo P., Girardi L., Salasnich B., Dal Cero C., Rubele S., Nanni A., 2012, MNRAS, 427, 127
Buxton M. M., Bailyn C. D., Capelo H. L., Chatterjee R., Dinçer T., Kalemci E., Tomsick J. A., 2012, AJ, 143, 130
Carretta E., Bragaglia A., Gratton R., D'Orazi V., Lucatello S., 2009, A&A, 508, 695
Chatterjee S., Rasio F. A., Sills A., Glebbeek E., 2013, ApJ, 777, 106
Dalessandro E., Salaris M., Ferraro F. R., Cassisi S., Lanzoni B., Rood R. T., Fusi Pecci F., Sabbi E., 2011, MNRAS, 410, 694
Degroote P., Conroy K., Hambleton K., Bloemen S., Pablo H., Giammarco J., Prša A., 2013, EAS Publ. Ser., 64, 277
de Vita R., Bertin G., Zocchi A., 2016, A&A, 590, A16
de Vita R., Trenti M., Bianchini P., Askar A., Giersz M., van de Ven G., 2017, MNRAS, 467, 4057
Fregeau J. M., Cheung P., Portegies Zwart S. F., Rasio F. A., 2004, MNRAS, 352, 1
Geller A. M., Leiner E. M., Chatterjee S., Leigh N. W. C., Mathieu R. D., Sills A., 2017, ApJ, 842, 1
Gieles M., Zocchi A., 2015, MNRAS, 454, 576
Giersz M., Heggie D. C., Hurley J. R., Hypki A., 2013, MNRAS, 431, 2184
Giersz M., Leigh N., Hypki A., Lützgendorf N., Askar A., 2015, MNRAS, 454, 3150
Guidi G., Scannapieco C., Walcher C. J., 2015, MNRAS, 454, 2381
Harris W. E., 1996, AJ, 112, 1487
Hénon M. H., 1971, Ap&SS, 14, 151
Hobbs G., Lorimer D. R., Lyne A. G., Kramer M., 2005, MNRAS, 360, 974
Hurley J., Tout C. A., 1998, MNRAS, 300, 977
Hurley J. R., Pols O. R., Tout C. A., 2000, MNRAS, 315, 543
Hurley J. R., Tout C. A., Pols O. R., 2002, MNRAS, 329, 897
Hut P., Cool A., Bailyn C., McMillan S., Livio M., Shara M., 2002, in Shara M. M., ed., ASP Conf. Ser. Vol. 263, Stellar Collisions, Mergers and their Consequences. Astron. Soc. Pac., San Francisco, p. 405
Hut P., McMillan S., Makino J., Portegies Zwart S., 2010, Astrophysics Source Code Library, record ascl:1010.076
Hypki A., Giersz M., 2017, MNRAS, 471, 2537
Kaluzny J., Pietrukowicz P., Thompson I. B., Krzeminski W., Schwarzenberg-Czerny A., Pych W., Stakhovsky G., 2005, MNRAS, 359, 677
King I. R., 1966, AJ, 71, 64
Kotulla R., Fritze U., Weilbacher P., Anders P., 2009, MNRAS, 396, 462
Kroupa P., 1995a, MNRAS, 277, 1507
Kroupa P., 1995b, MNRAS, 277, 1491
Kroupa P., Tout C. A., Gilmore G., 1993, MNRAS, 262, 545
Lamers H. J. G. L. M., Anders P., de Grijs R., 2006, A&A, 452, 131

Leigh N. W. C., Giersz M., Marks M., Webb J. J., Hypki A., Heinke C. O., Kroupa P., Sills A., 2015, MNRAS, 446, 226
Madrid J. P., Leigh N. W. C., Hurley J. R., Giersz M., 2017, MNRAS, 470, 1729
Marks M., Kroupa P., 2012, A&A, 543, A8
Milone A. P. et al., 2012, A&A, 540, A16
Pang X.-Y., Olczak C., Guo D.-F., Spurzem R., Kotulla R., 2016, Res. Astron. Astrophys., 16, 001
Pasquato M., Chung C., 2016, A&A, 589, A95
Peuten M., Zocchi A., Gieles M., Hénault-Brunet V., 2017, MNRAS, 470, 2736
Pols O. R., Schröder K.-P., Hurley J. R., Tout C. A., Eggleton P. P., 1998, MNRAS, 298, 525
Popescu B., Hanson M. M., 2009, AJ, 138, 1724
Popescu B., Hanson M. M., 2010, ApJ, 713, L21
Popescu B., Hanson M. M., 2014, ApJ, 780, 27
Prša A. et al., 2016, ApJS, 227, 29
Pych W., Kaluzny J., Krzeminski W., Schwarzenberg-Czerny A., Thompson I. B., 2001, A&A, 367, 148
Rozyczka M., Thompson I. B., Pych W., Narloch W., Poleski R., Schwarzenberg-Czerny A., 2017, Acta Astron., 67, 203
Sills A., Glebbeek E., Chatterjee S., Rasio F. A., 2013, ApJ, 777, 105
Sippel A. C., Hurley J. R., Madrid J. P., Harris W. E., 2012, MNRAS, 427, 167
Sollima A., Baumgardt H., Zocchi A., Balbinot E., Gieles M., Hénault-Brunet V., Varri A. L., 2015, MNRAS, 451, 2185
Sollima A., Dalessandro E., Beccari G., Pallanca C., 2017, MNRAS, 464, 3871
Spera M., Mapelli M., Bressan A., 2015, MNRAS, 451, 4086
Stetson P. B., 1987, PASP, 99, 191
Stetson P. B., 1994, PASP, 106, 250
Stodolkiewicz J. S., 1986, Acta Astron., 36, 19
Wang L., Spurzem R., Aarseth S., Nitadori K., Berczik P., Kouwenhoven M. B. N., Naab T., 2015, MNRAS, 450, 4070
Wang L. et al., 2016, MNRAS, 458, 1450
Zhuang Y., Zhang F., Anders P., Ruan Z., Cheng L., Kang X., 2015, MNRAS, 446, 4260

APPENDIX A: COCOA MANUAL

In this section we provide a brief manual that explains how to obtain and use the different modules of COCOA and also how option files for photometric procedures are created in the code.

A1 Requirements and installation

COCOA has been primarily developed in PYTHON 2.7. Subprograms in COCOA have been developed in C and FORTRAN. The code also requires the GNU Scientific Library to run the observation module. COCOA can be downloaded from the provided links.[8] After, extracting the downloaded file, there is a Makefile that needs to be run in order to compile the necessary subprograms to use COCOA on a UNIX machine. This can be done by using the following command in the terminal:

```
$ make
```

More detailed instructions for using COCOA are provided in the README.md file that comes with the code. We would like to request the users to appropriately acknowledge this paper in case they use COCOA or any of the subprograms and routines provided with it.

[8] https://github.com/abs2k12/COCOA or http://users.camk.edu.pl/askar/COCOA-17.tar.gz

A2 Projection module and input file

The projection module in COCOA (see Section 2.1) has been primarily designed to work with the snapshot that is output by the MOCCA code. The module reads each line of the snapshot that among other parameters provides the radial position, radial and tangential velocity of each single star, and the centre of mass position and velocities for each binary in the system. The module uses this data to obtain the projected position and velocity in x, y and z coordinates. The module also checks whether the object is a single star or a binary, in the case that the object is a binary star, the projected position of the individual stars in the binary with respect to the centre of the cluster is resolved. There is also an option to obtain the resolved individual velocities of the binary components with respect to the cluster centre or to have the centre of mass velocities for those binaries in the projected snapshot that is output by COCOA. In order to obtain the resolved position and velocities of individual stars in binary systems, we use the GET_ORBIT routine provided in the PHOEBE 2.0 code (Degroote et al. 2013; Prša et al. 2016) that primarily deals with physics of eclipsing binaries. The code solves the Kepler equation to determine the position and velocities of binary components at a given time. The description of the output produced by the projection module is provided in the README.md file.

With the projection module it is also possible to generate multiple snapshots that track the movement of the individual binary components in their orbits. The time for these snapshots can be defined by the user in units of days through two methods. Either they can provide a final time and the interval at which each snapshot should be generated or through an input file in which customized values can be provided by the user. In the future, in these multiple snapshots the magnitude of the binary components will be changed during eclipses and subsequent observations can be analysed to check for variability of those systems and detecting such binaries. The projection module also provides the possibility to generate a rotated projected snapshot. The rotation can be controlled by defining three angles that would define the three-dimensional rotation matrix.

The following input parameters need to be set in the input_projection.py file.

(i) snapshot_name – name of the snapshot file that will be projected.

(ii) seed – random number seed that is used to generate angles and distributions that are required for the project. The user can change the seed value to get different projections for the same snapshot.

(iii) binary_velocities – this parameter can take the value of either 1 or 0. If the value is set to 1, then in the output file, the projected individual velocities of stars in binaries are resolved with respect to the centre of the cluster. If this value is set to 0, then the projected centre of mass velocities of binary systems is provided in the output file.

(iv) intimes – this parameter can take the values 0, 1, and 2. When the value is set to 0, only one projected snapshot is generated. If the value is set to 1, then the user can generate multiple snapshots over short observational time that needs to be provided in days. With intimes set to 1, the user provides the end time and the time interval at which each snapshot should be generated. If intimes is set to 2, then the user can provide customized times after which each simulation is generated. To do this, the user needs to provide a data file in the working directory (time.dat) in which each row provides the time in days after which a snapshot will be generated.

(v) finaltimes – parameter is only relevant when intimes is set to 1. The value needs to be set for the final time in days at which the multiple snapshots should be generated.

(vi) interval – parameter is only relevant when intimes is set to 1. The value needs to be set for interval in days at which the multiple snapshots should be generated.

(vii) rotation – input parameter can take the value of either 0 or 1. In the case the value is 0, only the standard projected snapshot is given. If the value is 1, then a rotated projected snapshot is also provided.

(viii) alpha – parameter is only relevant when rotation is set to 1. The value of the angle needs to be input in units of degrees and this will define the rotation about the *x*-axis.

(ix) beta – similar to the alpha parameter, this parameter defines the rotation angle in degrees about the *y*-axis.

(x) gamma – defines the rotation angle in degrees about the *z*-axis.

There are additional input parameters in the input_projection.py file that can be enabled to call external routines that can generate surface brightness and velocity dispersion profiles from the simulation snapshot and fit the King (1966) model to those profiles. By default these options are disabled but can be enabled if the user requires them and comments have been added to the input file that explain how to enable these options. It should be noted that some of these external routines make use of functions taken from standard numerical recipes. Also the results from the fitting routines need to be carefully checked as one may encounter problems with the accuracy when fitting particular GC models.

Once the input_projection.py file has been correctly configured. The following command can be used to create the projected snapshot (or multiple snapshots depending on the parameters configured in the input file):

```
$ python projection.py
```

The output generated by the code will be the projected snapshot and a file called param_rtt.py that contains the name of the projected snapshot and the radial extent of the cluster in pc (this file is required by the imaging module).

A3 Imaging/observation module and SIM2OBS parameter files

Once the user has a projected snapshot, essentially three columns from it are needed to create FITS images of the synthetic observation. These are two columns for the projected *x*, *y* position of the star in the units of parsec and a third column containing the absolute magnitude in the filter/band for which the observations will be simulated. If the user already has a simulation snapshot with the projected position of the stars and their magnitudes, then they can skip the projection module and directly use the imaging/observation module by configuring the input_observation.py and param_rtt.py files.

The following input parameters are required when generating FITS images from a projected snapshot of a star cluster model and they can be configured by the user by editing the input_observation.py file. Below is a list of the input parameters in this file (variable names in the input file are shown in the parenthesis).

(i) NAXIS1 (valx) – image/CCD size in *x* (units of pixel, typical values 2048, 4096).

(ii) NAXIS2 (valy) – image/CCD size in *y* (units of pixel, typical values 2048, 4096).

(iii) DB-NX (posx) – *x*-coordinate column in projected snapshot file (value of *x* position should be in pc).

(iv) DB-NY (posy) – *y*-coordinate column in projected snapshot file (value of *y* position should be in pc).

(v) DB-NMAG (dmag) – magnitude column in projected snapshot file (absolute magnitude).

(vi) OBJECT (obje) – object name for which the mock observation is being generated (any string value may be entered).

(vii) DISTANCE (dist) – distance to the cluster in units of parsecs.

(viii) FILTER (filt) – filter name for which the simulation is being generated. Any string can be entered.

(ix) PIXSCALE (pixs) – gives the pixel scale of the instrument in units of arcsec pixel^{-1}. This defines the spatial resolution of the telescope.

(x) GAIN (gain) – instrument gain in units of photons ADU^{-1}.

(xi) SATLEVEL (satl) – saturation level of the detector in counts.

(xii) EXPOSURE (expo) – exposure for the observation as a multiple of direct counts.

(xiii) SEEING (seei) – seeing value for the observation in units of arcsec.

(xiv) BACKGROUND (back) – background level of the mock observation. Typical values between 1 and 10.

(xv) PSF (psff) – point spread function (PSF) model for the synthetic observation. Either string values 'G' or 'M' can be entered for this parameter. G for Gaussian profile and M for Moffat profile. There are no variations in the PSF produced by SIM2OBS.

(xvi) M_BETA (mbeta) – in the case a Moffat PSF is used, this defines the beta parameter for the Moffat distribution.

(xvii) NOISE (noise) – if set to 1, then this parameter introduces a Poisson noise to the observed FITS image. This parameter can take either the value 0 (no noise) or 1 (Poisson noise).

(xviii) RA_OFFSET (raof) – defines the offset value from the centre of the cluster in the *x* direction. This parameter is useful in imaging different field of the cluster model. Units for offset are in arcseconds.

(xix) DEC_OFFSET (deco) – defines the offset value from the centre of the cluster in the *y* direction. Units for offset are in arcseconds.

(xx) OVERLAP_FACTOR – this parameter defines the overlapping factor between adjoining frames when multiple images need to be made to image the entire cluster model. The value must always be less than 1.0. The default value for this parameter is set to 0.97, this means that there is a 3 per cent overlap between neighbouring frames.

There are two editable parameters in the param_rtt.py file. These are as follows.

(i) rtt – this defines the radial extent of the cluster till that the mock observations will be generated. If you would only like to image the cluster up to a certain radius, then you can enter that value for this parameter in units of parsec.

(ii) snapname – this variable should contain the name of the projected snapshot file in which you have the three columns that were specified in the input_observation.py file.

Once the input files have been configured. The user can generate the FITS images by running the following command in the terminal:

```
$ python observation.py
```

The code will calculate the number of images it needs to create to image the cluster depending on the input size of the CCD, pixel scale, and the radial extent of the cluster. All the images will be generated automatically. The output will be produced in the subdirectory 'fits-files'. The innermost frame will be named 0.0.fits, the naming convention for the output FITS files is based on the

coordinates of the grid of the mosaic. First digit corresponds to the *x*-position of the image in the grid of mosaics, and the second digit corresponds to the *y*-position of the image in that grid (x.y.fits).

A4 Photometry module

After all the FITS images required to image the cluster have been created. The user can use the photometry module to automatically carry out PSF photometry on each of the image. For the convenience of the user, the output of this routine will concatenate the photometry results from all individual images in the grid, clean the catalogue for overlapping stars in multiple images, and provide positions of the stars with respect to the cluster centre in units of arcseconds along with the instrumental magnitude and the error in the magnitude. In order to run the photometry module, you will need in the working directory of COCOA, compiled stand-alone versions of DAOPHOT and ALLSTAR (Stetson 1987, 1994). The module also requires the clean_overlap executable that is created in the working directory when the make command for COCOA is used. In order to run the photometry module the following commands needs to be used in the terminal:

```
$ python photometry.py
```

The module essentially creates wrappers that will then run DAOPHOT and ALLSTAR on the synthetic FITS images in order to carry out PSF photometry. The module will also generate the option files needed by these two codes based on the values of the FWHM of the synthetic observations. In SIM2OBS, the FWHM of the simulated image is defined by the seeing value divided by the pixel scale of the instrument (these parameters are defined in the imaging module, see Section A3). The prescriptions through which these option files are generated based on the FWHM value is shown in Section A5.

A5 Photometry options and parameter files

For daophot.opt, the following parameters are written in the file.

(i) FWHM. Taken to be the FWHM value determined by the stand-alone tool.

(ii) FIT = 1.2 × FWHM. This is the part of the PSF used to determine the fit to each star.

(iii) PSF = 2.7 × FWHM. PSF value defines the width of a square box inside which PSF will be determined.

(iv) READ = 0.1. This is the value for the readout noise that is used in our option files. Our simulated FITS images have no readout noise, the minimum value that is acceptable for DAOPHOT is used.

(v) GAIN. The value for gain is taken from the SIM2OBS parameter file that the user entered when generating the FITS images.

(vi) TH = 3.5. The threshold value is in the units of standard deviation of the sky noise and is kept at 3.5 or 5 in order to avoid reading the Poisson noise that is simulated in our images.

(vii) AN = 1. This parameter determines which analytical PSF model to generate. AN = 1 generates a Gaussian PSF model.

(viii) LOWBAD ∼ 10–15. Values between 10 and 15 are used for this parameter that is expressed in sky-sigma units. High value for this parameter prevents reading into the noise.

(ix) HIBAD. This value is also taken from the SIM2OBS parameter file. The user needs to enter the saturation level of the CCD when generating FITS image with SIM2OBS. The value entered for saturation level is used as the HIBAD value in the daophot.opt file.

(x) WATCH = −2.0. This suppresses detailed output to the terminal.

(xi) VAR=0. The synthetic observations do not have a variable PSF across the image, VAR is set to 0.

In the photo.opt, the apertures are defined. There is an option to define one or multiple apertures. For the results shown later in the document. A single aperture is defined.

(i) A1 = 1.5 × FWHM.

(ii) IS = 4.0 × FWHM. IS defines the inner radius of sky annulus.

(iii) OS = 5.0 × FWHM. OS defines the outer radius of sky annulus.

For the allstar.opt file, the following parameters are defined.

(i) fit = 1.2 × FWHM. Same value is used as for daophot.opt.

(ii) isky = 2.0 × FWHM. IS defines the inner radius of sky annulus in the allstar.opt file.

(iii) osky = 5.0 × FWHM. OS defines the outer radius of sky annulus.

(iv) watch = 0. This suppresses detailed output to the terminal.

(v) redet = 1. This parameter allows ALLSTAR to redetermine centroids of the star and to improve the quality of fit.

This paper has been typeset from a TEX/LATEX file prepared by the author.