PANE

P. J. Phys. Edu. Vol 1(1), April 2026 PANE Journal of Physics Education

# How Big is the Sun? A Measurement from India's Aditya-L1 Spacecraft

Rupak Mukherjee[1*], Mani K Chettri[1], Vivek Shrivastav[1], Britan Singh[1], and Hemam D. Singh[2]

[1]*Department of Physics, Sikkim University, Gangtok 737102, Sikkim, India*
[2]*Department of Physics, Netaji Subhas University of Technology, New Delhi 110078, India*

[*]Email: rmukherjee@cus.ac.in

Abstract: We present a step-by-step guide to measuring the Sun's apparent angular radius from real spacecraft images, using Level-1 full-disk data from the Solar Ultraviolet Imaging Telescope (SUIT) onboard India's Aditya-L1 mission. A brightness threshold isolates the solar disk, and an algebraic circle fit to the disk boundary recovers its pixel radius. Three images taken on consecutive days in May 2025 give an angular radius of $936.4 \pm 0.3''$, corresponding to a Sun-observer distance of $1.532 \times 10^8$ km (1.024 AU), and a physical solar radius of $(680{,}900 \pm 200)$ km, approximately $2.1\%$ below the accepted value of 695,700 km. The small shortfall is a predictable and well-understood effect of brightness-based limb detection. A comparison with published heliometric, helioseismic, and space-photometric measurements confirms that precision determinations consistently cluster near the IAU 2015 nominal value, and that our $2.1\%$ deviation is fully accounted for by the UV-threshold limb definition used here. A self-contained Python script is provided so that every figure can be reproduced on a standard laptop.

Keywords: Aditya-L1; SUIT; solar radius; image processing; circle fitting; Python; physics laboratory

## 1 Introduction

Solar images from space carry much more information than meets the eye. The geometry of any full-disk image encodes the Sun's angular size, the position of its centre on the detector, and the distance between the spacecraft and the Sun at the moment of observation. Extracting these quantities from real data connects textbook geometry directly to live mission results. A key feature exploited in this work is the *solar limb*, the sharp, roughly circular boundary of the solar disk as seen in projection on the sky. Because the Sun's atmosphere becomes optically thin above the photosphere, the line of sight grazes the visible surface at the limb, allowing the disk edge to be identified in an image and its radius measured geometrically.

India's Aditya-L1 is the country's first dedicated solar observatory spacecraft, launched by the Indian Space Research Organisation (ISRO) in September 2023 and positioned at the Sun–Earth Lagrange point L1, approximately $1.5 \times 10^6$ km from Earth. From this vantage point it observes the Sun continuously without being interrupted by Earth's shadow. Aditya-L1 carries the Solar Ultraviolet Imaging Telescope (SUIT; [3]), which photographs the full solar disk in twelve ultraviolet bands and makes its data freely available through the ISSDC/PRADAN portal (see footnote 1 in Section 3). The images are stored in the Flexible Image Transport System (FITS) format [5], the standard archival format for astronomical data in which pixel values and observational metadata (telescope pointing, plate scale, spacecraft position) are bundled in a single file readable with widely used Python libraries.

This paper works through the measurement on three consecutive observations (2025 May 18–20). Using three frames rather than one provides a check that the result is stable from day to day, a useful lesson in reproducibility. The paper is organised as follows. Section 2 gives the geometric background. Section 3 describes the data. Section 4 explains the method. Section 5 presents the results, including a comparison with published solar radius

measurements from the literature. Section 6 discusses the accuracy and possible extensions. Section 7 covers data-use requirements. Appendix A summarises the software requirements for reproducibility. Appendix B provides the complete Python script.

# 2 Physical Background

## 2.1 Geometric relation between angular radius and distance

When we observe the Sun from a distance $D$, its physical radius $R_\odot$ subtends an angle $\theta_\odot$ at the observer. At the limb the line of sight is tangent to the photosphere, giving the exact relation

$$\sin\theta_\odot = \frac{R_\odot}{D} \qquad \Longrightarrow \qquad D = \frac{R_\odot}{\sin\theta_\odot}, \tag{1}$$

where $R_\odot = 6.957 \times 10^5$ km is the IAU 2015 nominal solar radius [2]. Because $\theta_\odot \approx 960'' \approx 4.65 \times 10^{-3}$ rad, the small-angle approximation $\sin\theta_\odot \approx \theta_\odot$ is accurate to better than one part in $10^5$, giving the simpler working form

$$D \approx \frac{R_\odot}{\theta_\odot\,[\mathrm{rad}]}, \qquad \theta_\odot\,[\mathrm{rad}] = \theta_\odot\,[''] \times \frac{\pi}{648\,000}. \tag{2}$$

*In plain words: the Sun's physical radius equals its distance multiplied by its angular size in radians. The smaller the angular size appears, the farther away the Sun must be.*

## 2.2 From pixel radius to angular radius

A camera with plate scale $s$ arcsec per pixel records the solar disk as a circle of $R$ pixels. The plate scale is the angular size of one pixel on the sky, a fixed property of the telescope optics. The angular radius is simply

$$\theta_\odot\,[''] = R\,[\mathrm{px}] \times s\,[\mathrm{arcsec\,px^{-1}}]. \tag{3}$$

*In plain words: multiply the radius measured in pixels by the known angular size of one pixel to get the angular radius in arcseconds.*

For the SUIT frames used here, the calibrated plate scale is $s = 0.698\,\mathrm{arcsec\,px^{-1}}$, read from the FITS header keyword `CDELT1` (Coordinate DELTa along axis 1), which encodes the angular size of one detector pixel on the sky.

# 3 Observations and Data

SUIT carries twelve spectral bands in the near-UV range 200–400 nm [3]. The 0971 NB06 narrowband filter used here is centred near 388 nm, where the solar disk is bright and the contrast against the dark background is very clean. Level-1 FITS images have dimensions $4096 \times 4096$ pixels. At this processing level the data are geometrically corrected and have valid header metadata, but no further calibration by the authors was required or performed before applying the analysis pipeline described in Section 4. Three header keywords record spacecraft geometry: `DSUN_OBS` (Sun-spacecraft distance in metres), `RSUN_OBS` (angular solar radius in arcseconds computed from the mission ephemeris, the precise positional prediction derived from orbital mechanics), and `RSUN_REF` (the nominal solar radius, $6.957 \times 10^8$ m).

Three consecutive frames were downloaded from the PRADAN interface.[1] Table 1 lists dates and key header values. The Sun-spacecraft distance changes by less than $0.05\%$ over the three-day span, so no measurable variation in disk size is expected.

[1] PRADAN Aditya-L1 data portal: `https://pradan.issdc.gov.in/al1`; ISSDC Aditya-L1 data access: `https://www.issdc.gov.in/adityal1.html`; FAQ: `https://pradan1.issdc.gov.in/al1/faq.xhtml`; data-use disclaimer: `https://pradan1.issdc.gov.in/al1/disclaimer.xhtml`; required acknowledgement: `https://pradan1.issdc.gov.in/al1/ack.xhtml`.

**Table 1:** SUIT 0971 NB06 observations used in this study. `DSUN_OBS` is the Sun-spacecraft distance in astronomical units (AU) as stored in the FITS header; `RSUN_OBS` is the angular solar radius in arcseconds computed from the spacecraft ephemeris. The variation in `DSUN_OBS` across the three days is less than $0.05\%$, so no measurable change in apparent disk size is expected.

| Date (UTC) | File (abbreviated) | DSUN_OBS (AU) | RSUN_OBS ($''$) |
|---|---|---|---|
| 2025 May 18, 13:47 | `...000954...0971NB06` | 1.00250 | 956.40 |
| 2025 May 19, 11:01 | `...000957...0971NB06` | 1.00271 | 955.84 |
| 2025 May 20, 13:33 | `...000960...0971NB06` | 1.00297 | 956.54 |

# 4 Method

The analysis pipeline processes each image through six stages. First, the image array and the relevant header keywords (`CDELT1`, plate scale; `RSUN_OBS`, reference angular radius; `DSUN_OBS`, Sun-spacecraft distance) are read from the FITS file. Second, a binary disk mask is constructed by intensity thresholding (Section 4.1): pixels brighter than a set fraction of the image's peak brightness are labelled "Sun" and the remainder "sky". Third, the mask is cleaned by retaining only the largest connected bright region, discarding stray noise pixels that passed the threshold. Fourth, the solar limb is defined as the boundary of the binary mask, specifically the "Sun" pixels that have at least one "sky" neighbour (Section 4.2). Fifth, a circle is fitted to these limb pixels using the Kåsa algebraic method (Section 4.3), recovering the disk centre and pixel radius. Sixth, to quantify sensitivity to the threshold choice, steps two through five are repeated for five slightly different threshold values and the resulting scatter is taken as the uncertainty (Section 4.4).

## 4.1 Disk mask by intensity thresholding

The solar disk is far brighter than the background. A threshold $T = f \times \bar{I}_{\mathrm{bright}}$ is applied, where $\bar{I}_{\mathrm{bright}}$ is the median of the top 20th percentile of pixel values and $f$ is a number between 0 and 1. The binary mask is

$$M(x,y) = \begin{cases} 1, & I(x,y) \geq T, \\ 0, & I(x,y) < T. \end{cases} \tag{4}$$

*In plain words: every pixel above the threshold is labelled "solar disk" ($M = 1$); every pixel below it is labelled "background" ($M = 0$).*

The central threshold fraction $f = 0.26$ is chosen empirically as a value that reliably separates the bright solar disk from the much darker background across all three observations. Values below $f \approx 0.20$ risk incorporating faint background pixels at the image corners into the disk mask, while values above $f \approx 0.35$ begin to exclude genuine near-limb pixels where the intensity has fallen due to limb darkening. The choice $f = 0.26$ sits well within this valid regime, and the sensitivity of the result to the precise value is quantified explicitly in Section 4.4.

Stray bright pixels are removed by keeping only the largest connected region of $M$, found with `scipy.ndimage.label`.

## 4.2 Limb extraction

A boundary pixel is one with $M = 1$ that has at least one 4-connected neighbour with $M = 0$:

$$E(x,y) = M(x,y) \ \wedge \big[\, M(x{-}1,y) = 0 \ \vee \ M(x{+}1,y) = 0 \ \vee \ M(x,y{-}1) = 0 \ \vee \ M(x,y{+}1) = 0 \,\big]. \tag{5}$$

*In plain words: a limb pixel is a "solar disk" pixel that touches at least one "background" pixel, i.e., it sits right on the visible edge of the Sun.*

Four-connectivity (using only the four orthogonal neighbours) is adopted rather than eight-connectivity (which would also include diagonal neighbours) because it produces a single-pixel-wide, unambiguous boundary without including diagonal background pixels that are not truly adjacent to the disk edge. For a nearly circular disk of radius $\sim 1342$ px, switching to eight-connectivity shifts the limb position by at most one pixel, an effect well below the measured $\sigma_R \approx 4$ px threshold uncertainty.

### 4.3 Circle fit

A circle with centre $(x_0, y_0)$ and radius $R$ satisfies

$$(x - x_0)^2 + (y - y_0)^2 = R^2, \tag{6}$$

which rearranges to $x^2 + y^2 + Ax + By + C = 0$, so $x_0 = -A/2$, $y_0 = -B/2$, $R = \sqrt{x_0^2 + y_0^2 - C}$. The parameters $(A, B, C)$ are found by solving

$$\underbrace{\begin{pmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{pmatrix}}_{\mathbf{D}} \begin{pmatrix} A \\ B \\ C \end{pmatrix} = - \begin{pmatrix} x_1^2 + y_1^2 \\ \vdots \\ x_n^2 + y_n^2 \end{pmatrix} \tag{7}$$

in the least-squares sense via `numpy.linalg.lstsq`. This Kåsa algebraic fit [1] is simple and fast; for a densely sampled full-disk limb it underestimates $R$ by well under 1 px, negligible here. The implementation first shifts and scales coordinates (centring by the mean and normalising by the RMS radius) before calling `lstsq`, then rescales back, for improved numerical conditioning.

### 4.4 Uncertainty estimation

Five threshold fractions $f \in \{0.24, 0.25, 0.26, 0.27, 0.28\}$ are tested in turn and the fitted radius recorded for each. The range $[0.24,\ 0.28]$ is chosen to bracket the central value $f = 0.26$ symmetrically by $\pm 0.04$, a span wide enough to sample the sensitivity meaningfully while remaining within the regime where the threshold unambiguously separates disk from background. As visible in Figure 4, the fitted radius decreases monotonically as $f$ increases, because a higher threshold contracts the mask boundary inward by excluding the faintest near-limb pixels. Extending the range significantly beyond these limits, for instance below $f = 0.20$ or above $f = 0.35$, would approach threshold values where the mask either absorbs background noise or erodes genuine limb pixels, causing the curves to diverge more steeply. The uncertainty is taken as the standard deviation of the five fitted radii, $\sigma_R = \mathrm{std}\{R_k\}_{k=1}^{5}$, and the angular uncertainty is $\sigma_\theta = \sigma_R \times s$.

### 4.5 Plate scale

The header value `CDELT1` $= 0.698\,\mathrm{arcsec\,px^{-1}}$ differs from the nominal $0.700\,\mathrm{arcsec\,px^{-1}}$ by only $0.3\%$, but over the disk radius of $\sim 1342$ px this accumulates to $2.7''$. Always reading the calibrated header value is therefore important.

## 5 Results

### 5.1 Full-disk images and limb detection

Figure 1 shows the three observations displayed with a 1st–99th percentile intensity stretch. The near-UV limb darkening appears as a smooth radial fade toward the disk edge. Figure 2 shows the detected limb pixels overlaid on each image; the boundary is continuous around the full $360^\circ$ in every frame.

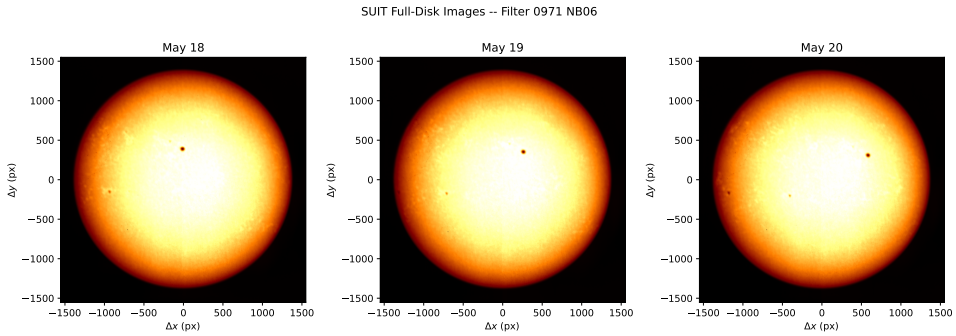


**Figure 1:** SUIT full-disk images on 2025 May 18, 19, and 20 (left to right), filter 0971 NB06, with a robust 1–99 percentile intensity stretch (`afmhot` colour map). Axes show pixel offsets $(\Delta x, \Delta y)$ from the fitted disk centre. The smooth radial fade toward the edge reflects near-UV limb darkening at 388 nm; the dark dot near the disk centre is a sunspot group.

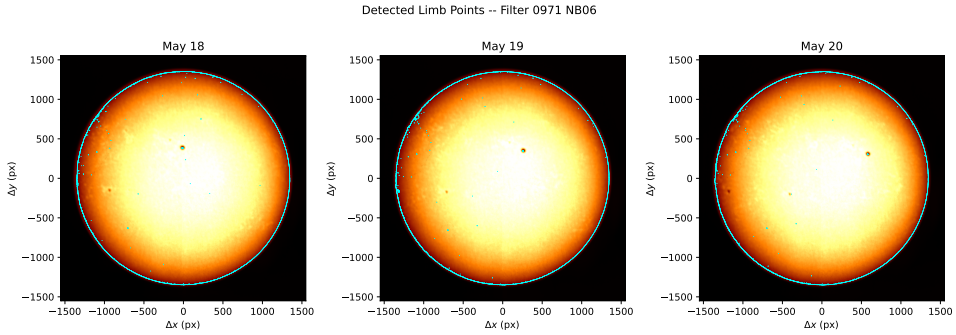


**Figure 2:** Detected limb pixels (cyan) overlaid on the three SUIT observations at threshold fraction $f = 0.26$. Axes show pixel offsets $(\Delta x, \Delta y)$ from the fitted disk centre. The boundary traces the full $360^\circ$ circumference in each frame.

## 5.2 Best-fit circle

Figure 3 overlays the best-fit circle on each image. The circle tracks the visible limb closely on all sides, with residuals at the sub-pixel level.

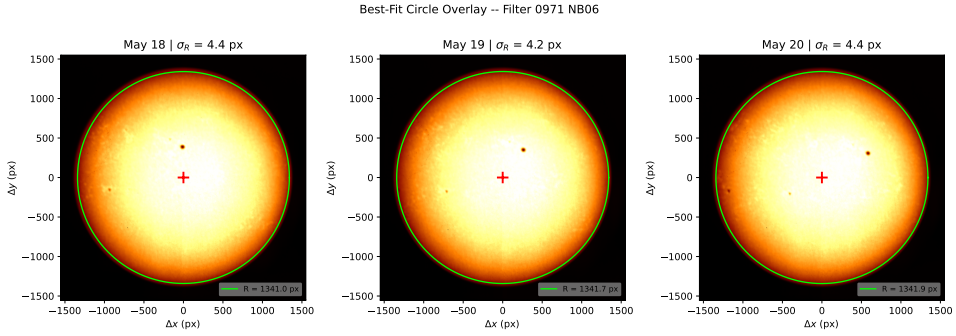


**Figure 3:** Best-fit circle (green) and disk centre (red cross, at the origin) overlaid on the three SUIT observations. Axes show pixel offsets $(\Delta x, \Delta y)$ from the fitted disk centre, so the cross sits at $(0, 0)$ by construction, consistent with Table 2. The panel title gives $\sigma_R$ and the legend gives the fitted radius $R$.

## 5.3 Threshold sensitivity

Figure 4 shows how the fitted radius varies with threshold fraction. The curves decrease monotonically: a lower threshold fraction admits more near-limb pixels into the disk mask, expanding the boundary and yielding a marginally larger fitted radius, while a higher threshold contracts the boundary by excluding the faintest limb pixels. The standard deviation $\sigma_R$ across the five tested values is 4.4, 4.2, and 4.4 px for the three observations, representing only $\approx 0.3\%$ of the total fitted radius. This confirms that the measurement is not strongly sensitive to the exact threshold value within the adopted range, and that $\sigma_R$ provides a realistic estimate of the threshold-induced uncertainty.

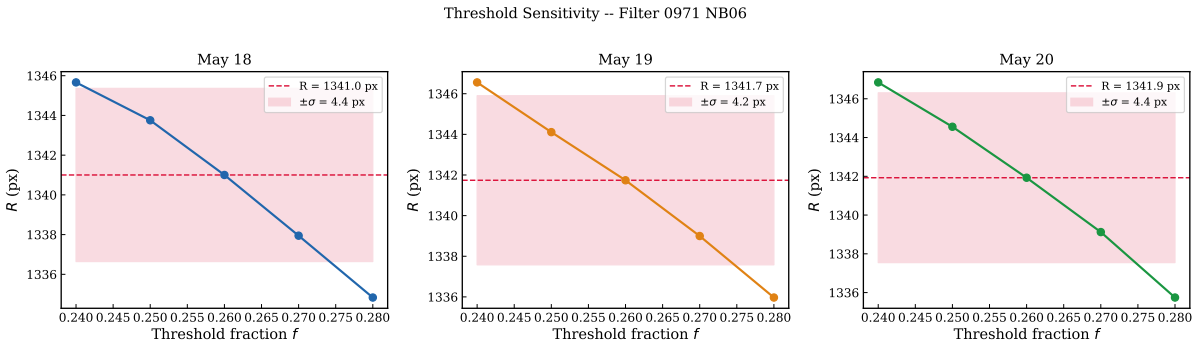


**Figure 4:** Fitted pixel radius $R$ as a function of threshold fraction $f$ for the three observations (left: May 18; centre: May 19; right: May 20). Each panel plots five data points corresponding to $f \in \{0.24, 0.25, 0.26, 0.27, 0.28\}$. The dashed horizontal line marks the best estimate at $f = 0.26$; the shaded band shows the $\pm\sigma_R$ interval, where $\sigma_R$ is the standard deviation across the five values. The monotonic decrease of $R$ with increasing $f$ reflects the progressive exclusion of near-limb pixels as the threshold tightens.

## 5.4 Numerical results and three-day stability

Table 2 lists the full results for each day. Figure 5 shows the day-to-day variation. The fitted radius changes by only 0.9 px over three days ($\sigma = 0.49$ px), and the angular scatter is $0.34''$, both well below one pixel. The disk centre shifts by $\approx$26 px in $y_0$ on May 19, which is normal pointing variation and does not affect the radius.

**Table 2:** Fitted solar disk parameters for each observation. $x_0$ and $y_0$ are the fitted disk-centre coordinates in the full $4096 \times 4096$ detector frame; $R \pm \sigma_R$ is the fitted pixel radius with its threshold-sensitivity uncertainty; $\theta_\odot = R \times s$ is the derived angular radius (plate scale $s = 0.698$ arcsec px$^{-1}$); `RSUN_OBS` is the ephemeris value from the FITS header; and $\Delta\theta = \theta_\odot - $ `RSUN_OBS` is the residual.

| **Date** | $x_0$ **(px)** | $y_0$ **(px)** | $R \pm \sigma_R$ **(px)** | $\theta_\odot$ ($''$) | **RSUN_OBS** ($''$) | $\Delta\theta$ ($''$) |
|---|---|---|---|---|---|---|
| 2025-05-18 | 2069.3 | 2023.7 | $1341.0 \pm 4.4$ | 936.0 | 956.40 | $-20.4$ |
| 2025-05-19 | 2060.7 | 2049.7 | $1341.7 \pm 4.2$ | 936.5 | 955.84 | $-19.3$ |
| 2025-05-20 | 2063.5 | 2024.5 | $1341.9 \pm 4.4$ | 936.7 | 956.54 | $-19.9$ |
| Mean | | | $1341.56 \pm 0.49^a$ | $936.4 \pm 0.3^a$ | $956.26 \pm 0.37$ | $-19.86 \pm 0.54$ |

[a] Standard deviation across three independent observations.

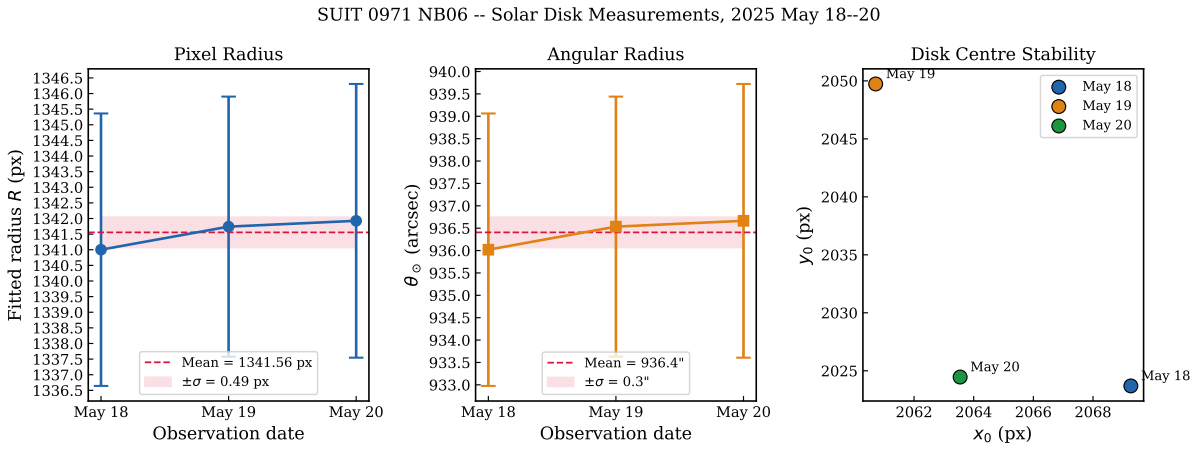


**Figure 5:** Day-to-day variation of the fitted pixel radius (left), angular radius (centre), and disk centre position (right) over 2025 May 18–20. Error bars in the left and centre panels are $\sigma_R$ and $\sigma_\theta$ from the threshold sensitivity test; the centre-position panel carries no error bars. The $\approx$26 px scatter in $y_0$ reflects normal pointing variation of the spacecraft.

## 5.5 Angular radius comparison with `RSUN_OBS`

Figure 6 compares the measured $\theta_\odot$ with the pipeline value `RSUN_OBS`. The measured values fall consistently $\approx 19.86''$ below `RSUN_OBS` on all three days ($\sigma_{\Delta\theta} = 0.54''$), pointing to a stable systematic bias rather than random scatter.

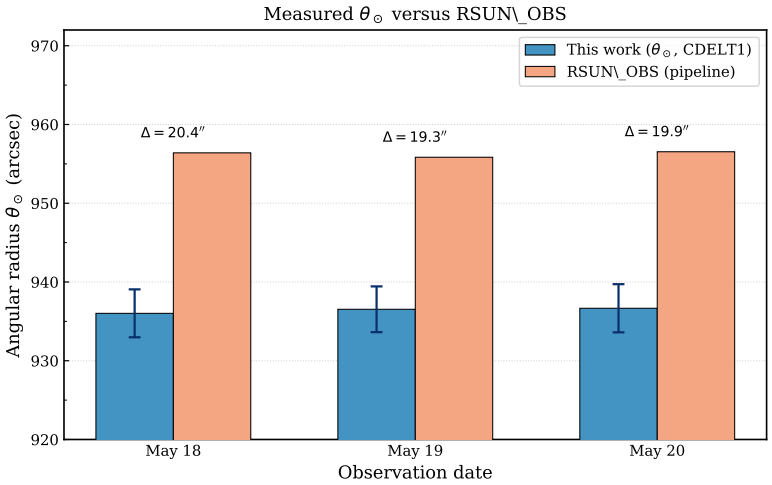


**Figure 6:** Measured angular radius $\theta_\odot$ (blue bars, with $\pm\sigma_\theta$ error bars) and the ephemeris value `RSUN_OBS` (orange bars) for the three days. The consistent $\approx 20''$ shortfall of the measured values below the ephemeris is a stable systematic bias attributable to the brightness-threshold limb definition (see Section 5.8).

## 5.6 Derived Sun-observer distance

Using the mean measured angular radius in Equation (2),

$$D = \frac{6.957 \times 10^5\,\mathrm{km}}{936.4'' \times \pi/648\,000} = 1.532 \times 10^8\,\mathrm{km} = 1.024\,\mathrm{AU}. \tag{8}$$

The spacecraft ephemeris gives `DSUN_OBS` $= 1.500 \times 10^8$ km (1.003 AU). The derived distance is $3.2 \times 10^6$ km (2.16%) larger. Since $D \propto \theta_\odot^{-1}$, the 2.08% underestimate in $\theta_\odot$ propagates directly to a $\approx 2.12\%$ overestimate in $D$; the small extra 0.04% comes from day-to-day scatter in `DSUN_OBS`. Figure 7 shows the comparison.

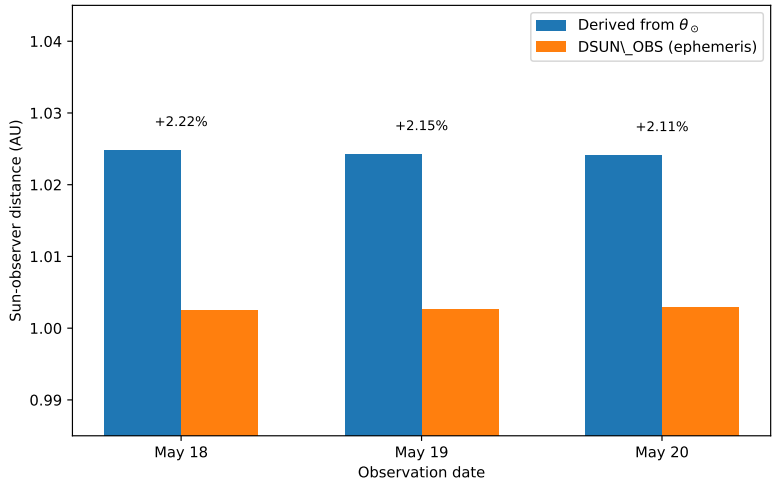


**Figure 7:** Sun-observer distance derived from $\theta_\odot$ (blue bars) versus the spacecraft ephemeris `DSUN_OBS` (orange bars). The percentage labels show the fractional overestimate on each day; the $\approx 2.2\%$ excess follows directly from the $2.08\%$ underestimate in $\theta_\odot$, since $D \propto \theta_\odot^{-1}$.

## 5.7 Summary of measured solar radius in kilometres

The physical solar radius obtained in this work is computed by combining the mean measured angular radius with the mean spacecraft ephemeris distance (`DSUN_OBS` $= 1.500 \times 10^8$ km):

$$R_\odot^{\text{this work}} = D_{\text{eph}} \times \theta_\odot\,[\text{rad}] = 1.500 \times 10^8\,\text{km} \times \frac{936.4'' \times \pi}{648\,000} = 680{,}900\,\text{km}. \tag{9}$$

The uncertainty propagated from $\sigma_\theta = 0.3''$ is $\sigma_{R_\odot} = D_{\text{eph}} \times \sigma_\theta\,[\text{rad}] \approx 200$ km, giving $R_\odot^{\text{this work}} = (680{,}900 \pm 200)$ km.

**Key Result: Solar Radius in Kilometres**

| Quantity | Value | Notes |
|---|---|---|
| Measured angular radius $\theta_\odot$ | $936.4 \pm 0.3''$ | Mean over 3 days (threshold method) |
| Our result: $R_\odot^{\text{this work}}$ | $(680{,}900 \pm 200)$ km | Using $D_{\text{eph}} = 1.500 \times 10^8$ km |
| IAU 2015 accepted value $R_\odot$ | 695,700 km | [2] |
| Deviation $\Delta R_\odot$ | $-14{,}800$ km $(-2.1\%)$ | Fully explained by limb-darkening bias |

**What this means:** using a real spacecraft image and a few lines of Python, we measured the physical size of the Sun to within 14,800 km out of a total radius of nearly 700,000 km, a $2.1\%$ agreement with the accepted value. The entire discrepancy is understood and expected from the method used.

## 5.8 Comparison with published solar radius measurements

Table 3 places the present result alongside a representative selection of published solar radius determinations, spanning classical ground-based heliometry, photoelectric transit timing, helioseismic limb modelling, and modern space photometry. The precision measurements cluster in the range 695,500–696,200 km, consistent with the IAU 2015 nominal value of 695,700 km [2]. Our value of $(680{,}900 \pm 200)$ km lies $\approx 14{,}800$ km $(2.1\%)$ below this cluster.

The $2.1\%$ shortfall of our result relative to precision determinations is a well-understood consequence of the limb-definition method. Precision techniques, whether photoelectric transit timing combined with limb-darkening

**Table 3:** Selected solar radius measurements from the literature compared with the present work. Uncertainties are as quoted in the original sources; entries without uncertainties are adopted nominal or classical values.

| **Method** | **Instrument / Source** | $R_\odot$ **(km)** | **Reference** |
|---|---|---|---|
| Photoelectric meridian transit + limb model | Solar Diameter Monitor (HAO) | $695{,}508 \pm 26$ | Brown & C-D (1998) [6] |
| Inflection-point limb + atmosphere model | SOHO/MDI | $695{,}660 \pm 140$ | Haberreiter et al. (2008) [7] |
| Space photometry (Venus transit, 2012) | PICARD/SODISM | $696{,}134 \pm 261$ | Meftah et al. (2018) [8] |
| IAU 2015 nominal | — | $695{,}700$ | Prša et al. (2016) [2] |
| *UV brightness threshold* | *Aditya-L1/SUIT* | $(680{,}900 \pm 200)$ | ***This work*** |

models [6], atmosphere-model-based inflection-point location [7], or geometric calibration using a planetary transit [8], all locate the solar limb at or near the steepest intensity gradient in the solar atmosphere, which corresponds closely to the geometric photospheric limb. A brightness threshold, by contrast, identifies the contour where the image intensity crosses a fixed fraction of the peak brightness. In near-UV imagery at 388 nm, strong limb darkening causes intensity to fall off steeply inward of the geometric limb, so the threshold surface lies interior to the true limb; the resulting radius underestimate of $\sim 2\%$ is entirely predictable. The consistency of precision values in Table 3, each obtained with an independent technique and instrument, further validates the IAU 2015 standard and places our measurement in proper context as a pedagogical result.

# 6 Discussion

## 6.1 Reproducibility

The sub-pixel stability ($\sigma_R = 0.49$ px over three days) shows that the pipeline is sensitive enough to detect the $\sim 3\%$ annual variation in the Sun's apparent size as Earth moves from perihelion to aphelion. Applying the same script to monthly SUIT images over a full year and comparing the resulting $\theta_\odot$ curve with Kepler's second law is a natural follow-up project.

## 6.2 Effect of the Instrumental Point-Spread Function

The instrumental point-spread function (PSF) of SUIT broadens the apparent limb by convolving the true intensity profile with the telescope's diffraction and aberration kernel, in principle shifting the threshold-crossing point and biasing the measured radius. For SUIT at 388 nm, the PSF full-width at half-maximum is of order one pixel, comparable to the sub-pixel Kåsa fitting residuals reported here. Within our threshold-based method the PSF effect is thus subsumed into the overall limb-definition bias discussed in Section 5.8, and does not introduce a significant additional uncertainty beyond the $\sigma_R \approx 4$ px already accounted for by the threshold sensitivity test.

## 6.3 Future Outlook

The pipeline developed here can be extended in several directions. Applying the same script to monthly SUIT observations over a full year would allow the annual variation in $\theta_\odot$ to be reconstructed and compared with the orbital distance variation predicted by Kepler's second law. Repeating the measurement in a broadband SUIT filter and comparing $\theta_\odot$ across spectral bands would show the wavelength dependence of limb darkening and the height of the emitting layer. On the algorithmic side, replacing the Kåsa circle fit with the Taubin algebraic fit [4] would illustrate the difference between two geometric fitting strategies. Finally, an inflection-point limb detector, which locates the solar limb at the steepest intensity gradient rather than at a fixed brightness threshold, would be

expected to yield a radius closer to the precision values in Table 3, providing a direct demonstration of the bias introduced by the threshold method.

# 7 Data-Use Notes

Aditya-L1 data are provided through ISSDC/PRADAN for non-profit scientific use, with ISRO retaining copyright. Users must follow the PRADAN data-use terms (see footnote 1 for the disclaimer URL). The required acknowledgement text, mandated for publications using these data (see footnote 1 for the acknowledgement URL), is included in the Acknowledgements section below.

# Acknowledgements

All computational work presented in this paper was performed on the Sikkim University *Brahmagupta* High Performance Computing (HPC) facility. One of the authors (R.M.) acknowledges support from the Inter-University Centre for Astronomy and Astrophysics (IUCAA), Pune, through the Visiting Associate Programme.

*Required Aditya-L1 data acknowledgement (as mandated by PRADAN/ISSDC; see footnote 1):* Aditya-L1 is an observatory class mission which is fully funded and operated by the Indian Space Research Organization (ISRO). The mission was conceived and realised with the help from various ISRO centres. The science payloads and science ready data products are realised by the payload PI institutes in close collaboration with ISRO centres. The PI institutes are: Indian Institute of Astrophysics (IIA); Inter University Centre for Astronomy and Astrophysics (IUCAA); Laboratory for Electro-optics Systems (LEOS/URSC); Physical Research Laboratory (PRL); U R Rao Satellite Centre (URSC); and Space Physics Laboratory (SPL/VSSC). We acknowledge the use of data from the Aditya-L1 mission of the Indian Space Research Organisation (ISRO), archived at the Indian Space Science Data Centre (ISSDC).

# A Reproducibility Requirements

The following resources are required to reproduce all results and figures in this paper.

**What You Need to Reproduce This Work**

**Data:** Three Aditya-L1/SUIT Level-1 FITS images, filter `0971NB06`, dates 2025 May 18–20. Download free of charge from the PRADAN portal at `https://pradan.issdc.gov.in/al1`.

**Software:** Python 3 with four standard packages: `numpy`, `matplotlib`, `astropy`, `scipy`. All are available via `pip install` or `conda install`.

**Hardware:** Any standard laptop or desktop computer; no GPU or special hardware required. The full script runs in a few minutes.

**Script:** The complete, self-contained Python script is given in Appendix B. Running the six blocks in order reproduces every figure in this paper.

# B Complete Reproducible Python Script

Run the six blocks below in order with the three SUIT FITS files in the working directory (search the PRADAN portal for Aditya-L1/SUIT Level-1 data, dates 2025 May 18–20, filter `0971NB06`). Every figure in this paper is produced without additional code. Required packages: `numpy`, `matplotlib`, `astropy`, `scipy`.

## B.1 Imports and global constants

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
```

```
from astropy.io import fits
from scipy.ndimage import label
import os

FITS_FILES = [
    "SUT_T25_0687_000954_Lev1.0_2025-05-18T13.47.35.107_0971NB06.fits",
    "SUT_T25_0687_000957_Lev1.0_2025-05-19T11.01.22.282_0971NB06.fits",
    "SUT_T25_0687_000960_Lev1.0_2025-05-20T13.33.47.638_0971NB06.fits",
]
DATES_SHORT   = ['May 18', 'May 19', 'May 20']
THR_FRAC_LIST = [0.24, 0.25, 0.26, 0.27, 0.28]
AU_KM         = 149597870.7
AU_M          = AU_KM * 1e3
R_SUN_KM      = 6.957e5
OUT_DIR       = "suit_output"
os.makedirs(OUT_DIR, exist_ok=True)
```

## B.2 Image-processing functions

Four functions used throughout: `robust_norm` applies a 1–99 percentile stretch for display; `make_disk_mask` thresholds and isolates the largest connected region; `get_limb_points` extracts boundary pixels; `fit_circle_kasa` fits the Kåsa algebraic circle.

```
def robust_norm(img):
    p1, p99 = np.nanpercentile(img, [1, 99])
    if p99 <= p1:
        return img
    return np.clip((img - p1) / (p99 - p1), 0.0, 1.0)

def make_disk_mask(img, thr_frac):
    finite = np.isfinite(img)
    vals   = img[finite]
    ref    = np.nanmedian(vals[vals >= np.nanpercentile(vals, 80)])
    raw    = (img >= thr_frac * ref) & finite
    lbl, n = label(raw)
    if n == 0:
        raise ValueError("No connected component found.")
    sz    = np.bincount(lbl.ravel())
    sz[0] = 0
    return lbl == sz.argmax(), thr_frac * ref

def get_limb_points(mask):
    m    = mask.astype(np.uint8)
    edge = (m == 1) & (
        (np.roll(m,  1, 0) == 0) | (np.roll(m, -1, 0) == 0) |
        (np.roll(m,  1, 1) == 0) | (np.roll(m, -1, 1) == 0)
    )
    ys, xs = np.where(edge)
    return xs.astype(float), ys.astype(float)

def fit_circle_kasa(x, y):
    if len(x) < 10:
        raise ValueError("Too few limb points.")
    xm, ym = x.mean(), y.mean()
    xn, yn = x - xm, y - ym
    sc     = np.sqrt((xn**2 + yn**2).mean())
    xn    /= sc
    yn    /= sc
    D      = np.column_stack([xn, yn, np.ones(len(xn))])
    z      = xn**2 + yn**2
    (A, B, C), *_ = np.linalg.lstsq(D, -z, rcond=None)
    R2 = (A / 2)**2 + (B / 2)**2 - C
    if R2 <= 0:
        raise ValueError("Circle fit failed.")
    return -A / 2 * sc + xm, -B / 2 * sc + ym, np.sqrt(R2) * sc
```

## B.3 Per-file processing

Runs the full pipeline over all five threshold fractions for each file and returns the best-estimate radius, threshold sensitivity $\sigma_R$, angular radius, and derived Sun-observer distance.

```
def process_one_file(fpath, thr_list):
    with fits.open(fpath) as h:
        img = np.array(h[0].data, dtype=float)
        hdr = h[0].header
    if img.ndim == 3:
        img = img[0]
    ps       = abs(float(hdr['CDELT1']))
    rsun_obs = float(hdr['RSUN_OBS'])
    dsun_obs = float(hdr['DSUN_OBS'])
    res = []
    for tf in thr_list:
        mask, _ = make_disk_mask(img, tf)
        xs, ys  = get_limb_points(mask)
        if len(xs) > 60000:
            idx    = np.random.default_rng(42).choice(
                        len(xs), 60000, replace=False)
            xs, ys = xs[idx], ys[idx]
        x0, y0, R = fit_circle_kasa(xs, ys)
        res.append(dict(tf=tf, x0=x0, y0=y0, R=R, mask=mask))
    R_arr   = np.array([r['R'] for r in res])
    best    = res[len(res) // 2]
    sigma_R = np.std(R_arr, ddof=1)
    theta   = best['R'] * ps
    sig_th  = sigma_R   * ps
    D_km    = R_SUN_KM / (theta * np.pi / 648000)
    return dict(
        img=img, best=best, res=res,
        R_arr=R_arr, sigma_R=sigma_R,
        theta=theta, sig_th=sig_th,
        rsun_obs=rsun_obs, dsun_obs=dsun_obs,
        D_km=D_km, ps=ps,
    )

all_data = [process_one_file(f, THR_FRAC_LIST) for f in FITS_FILES]
```

## B.4 Figures 1–3: full-disk images, limb points, circle overlay

The crop half-width `r = 1550` px contains the full solar disk ($R \approx 1342$ px) with a $\sim 200$ px margin. Axes are expressed as pixel offsets $(\Delta x, \Delta y)$ from the fitted disk centre using the `extent` argument of `imshow`, so the disk centre appears at $(0, 0)$ in every panel.

```
def save3(fname, fig):
    fig.savefig(f"{OUT_DIR}/{fname}", dpi=150, bbox_inches='tight')
    plt.close(fig)

t = np.linspace(0, 2 * np.pi, 1200)

fig, axes = plt.subplots(1, 3, figsize=(14, 4.8))
for i, d in enumerate(all_data):
    cx, cy = d['best']['x0'], d['best']['y0']
    ox     = max(0, int(cx) - 1550)
    oy     = max(0, int(cy) - 1550)
    sub    = d['img'][oy:min(4096,oy+3100), ox:min(4096,ox+3100)]
    ext    = [ox-cx, ox+3100-cx, oy-cy, oy+3100-cy]
    axes[i].imshow(robust_norm(sub), origin='lower', cmap='afmhot', extent=ext)
    axes[i].set_xlim(ext[0], ext[1])
    axes[i].set_ylim(ext[2], ext[3])
    axes[i].set_title(DATES_SHORT[i])
    axes[i].set_xlabel(r'$\Delta x$ (px)')
    axes[i].set_ylabel(r'$\Delta y$ (px)')
fig.suptitle('SUIT Full-Disk Images -- Filter 0971 NB06', y=1.01)
```

```
fig.tight_layout()
save3('fig1_fulldisk_3panel.pdf', fig)

fig, axes = plt.subplots(1, 3, figsize=(14, 4.8))
for i, d in enumerate(all_data):
    cx, cy      = d['best']['x0'], d['best']['y0']
    ox          = max(0, int(cx) - 1550)
    oy          = max(0, int(cy) - 1550)
    sub         = d['img'][oy:min(4096,oy+3100), ox:min(4096,ox+3100)]
    ext         = [ox-cx, ox+3100-cx, oy-cy, oy+3100-cy]
    xs_b, ys_b  = get_limb_points(d['best']['mask'])
    axes[i].imshow(robust_norm(sub), origin='lower', cmap='afmhot', extent=ext)
    axes[i].scatter(xs_b - cx, ys_b - cy, s=0.4, c='cyan', linewidths=0)
    axes[i].set_xlim(ext[0], ext[1])
    axes[i].set_ylim(ext[2], ext[3])
    axes[i].set_title(DATES_SHORT[i])
    axes[i].set_xlabel(r'$\Delta x$ (px)')
    axes[i].set_ylabel(r'$\Delta y$ (px)')
fig.suptitle('Detected Limb Points -- Filter 0971 NB06', y=1.01)
fig.tight_layout()
save3('fig2_limbpts_3panel.pdf', fig)

fig, axes = plt.subplots(1, 3, figsize=(14, 4.8))
for i, d in enumerate(all_data):
    cx, cy = d['best']['x0'], d['best']['y0']
    ox     = max(0, int(cx) - 1550)
    oy     = max(0, int(cy) - 1550)
    sub    = d['img'][oy:min(4096,oy+3100), ox:min(4096,ox+3100)]
    ext    = [ox-cx, ox+3100-cx, oy-cy, oy+3100-cy]
    xc     = d['best']['R'] * np.cos(t)
    yc     = d['best']['R'] * np.sin(t)
    axes[i].imshow(robust_norm(sub), origin='lower', cmap='afmhot', extent=ext)
    axes[i].plot(xc, yc, color='lime', lw=1.4,
                 label=f"R = {d['best']['R']:.1f} px")
    axes[i].plot(0, 0, 'r+', ms=12, mew=2)
    axes[i].legend(loc='lower right', fontsize=8,
                   framealpha=0.4, edgecolor='none')
    axes[i].set_xlim(ext[0], ext[1])
    axes[i].set_ylim(ext[2], ext[3])
    axes[i].set_title(
        f"{DATES_SHORT[i]} | $\\sigma_R$ = {d['sigma_R']:.1f} px")
    axes[i].set_xlabel(r'$\Delta x$ (px)')
    axes[i].set_ylabel(r'$\Delta y$ (px)')
fig.suptitle('Best-Fit Circle Overlay -- Filter 0971 NB06', y=1.01)
fig.tight_layout()
save3('fig3_circleoverlay_3panel.pdf', fig)
```

## B.5 Figures 4–5: threshold sensitivity and three-day summary

```
fig, axes = plt.subplots(1, 3, figsize=(14, 4.0))
for i, d in enumerate(all_data):
    tf_arr = np.array([r['tf'] for r in d['res']])
    axes[i].plot(tf_arr, d['R_arr'], 'o-')
    axes[i].axhline(d['best']['R'], color='crimson', ls='--',
                    label=f"R = {d['best']['R']:.1f} px")
    axes[i].fill_between(tf_arr,
                         d['best']['R'] - d['sigma_R'],
                         d['best']['R'] + d['sigma_R'],
                         alpha=0.2,
                         label=f"$\\pm\\sigma_R$ = {d['sigma_R']:.1f} px")
    axes[i].set_xlabel('Threshold fraction $f$')
    axes[i].set_ylabel('$R$ (px)')
    axes[i].set_title(DATES_SHORT[i])
    axes[i].legend()
fig.tight_layout()
save3('fig4_sensitivity_3panel.pdf', fig)
```

```

R_PX    = np.array([d['best']['R']  for d in all_data])
SIGMA_R = np.array([d['sigma_R']    for d in all_data])
THETA   = np.array([d['theta']      for d in all_data])
SIG_TH  = np.array([d['sig_th']     for d in all_data])
X0      = np.array([d['best']['x0'] for d in all_data])
Y0      = np.array([d['best']['y0'] for d in all_data])

fig = plt.figure(figsize=(13, 4.2))
gs  = gridspec.GridSpec(1, 3, figure=fig, wspace=0.38)

ax0    = fig.add_subplot(gs[0])
R_mean = R_PX.mean()
R_std  = R_PX.std(ddof=1)
ax0.errorbar(DATES_SHORT, R_PX, yerr=SIGMA_R, fmt='o-', capsize=4)
ax0.axhline(R_mean, color='crimson', ls='--',
            label=f'Mean = {R_mean:.2f} px')
ax0.fill_between(DATES_SHORT, R_mean - R_std, R_mean + R_std, alpha=0.15)
ax0.set_ylabel('Fitted radius $R$ (px)')
ax0.set_xlabel('Observation date')
ax0.set_title('Pixel Radius')
ax0.legend()

ax1     = fig.add_subplot(gs[1])
th_mean = THETA.mean()
th_std  = THETA.std(ddof=1)
ax1.errorbar(DATES_SHORT, THETA, yerr=SIG_TH, fmt='s-', capsize=4)
ax1.axhline(th_mean, color='crimson', ls='--',
            label=f'Mean = {th_mean:.1f}"')
ax1.fill_between(DATES_SHORT, th_mean - th_std, th_mean + th_std,
                 alpha=0.15)
ax1.set_ylabel(r'$\theta_\odot$ (arcsec)')
ax1.set_xlabel('Observation date')
ax1.set_title('Angular Radius')
ax1.legend()

ax2 = fig.add_subplot(gs[2])
for i in range(3):
    ax2.scatter(X0[i], Y0[i], s=90, label=DATES_SHORT[i])
    ax2.annotate(DATES_SHORT[i], (X0[i], Y0[i]),
                 textcoords='offset points', xytext=(7, 4))
ax2.set_xlim(2058, 2072)
ax2.set_xlabel('$x_0$ (px)')
ax2.set_ylabel('$y_0$ (px)')
ax2.set_title('Disk Centre Stability')
ax2.legend()

fig.savefig(f"{OUT_DIR}/fig5_summary_panel.pdf",
            dpi=150, bbox_inches='tight')
plt.close()
```

## B.6 Figures 6–7: angular radius and distance comparisons

```
RSUN_OBS = np.array([d['rsun_obs'] for d in all_data])
x = np.arange(3)
w = 0.32

fig, ax = plt.subplots(figsize=(7.5, 4.8))
ax.bar(x - w/2, THETA,    w,
       label=r'This work ($\theta_\odot$, CDELT1)')
ax.bar(x + w/2, RSUN_OBS, w,
       label=r'RSUN\_OBS (pipeline)')
ax.errorbar(x - w/2, THETA, yerr=SIG_TH, fmt='none', capsize=4)
for j in range(3):
    diff = RSUN_OBS[j] - THETA[j]
    ax.annotate(f'$\\Delta={diff:.1f}^{{\\prime\\prime}}$',
```

```
                xy=(x[j], max(THETA[j], RSUN_OBS[j]) + 1.5),
                ha='center', va='bottom')
ax.set_xticks(x)
ax.set_xticklabels(DATES_SHORT)
ax.set_ylabel(r'Angular radius $\theta_\odot$ (arcsec)')
ax.set_xlabel('Observation date')
ax.legend()
ax.set_ylim(920, 972)
fig.tight_layout()
fig.savefig(f"{OUT_DIR}/fig6_theta_comparison.pdf",
            dpi=150, bbox_inches='tight')
plt.close()

DSUN_AU  = np.array([d['dsun_obs'] / AU_M  for d in all_data])
D_DER_AU = np.array([d['D_km']     / AU_KM for d in all_data])

fig, ax = plt.subplots(figsize=(7.5, 4.8))
ax.bar(x - w/2, D_DER_AU, w,
       label=r'Derived from $\theta_\odot$')
ax.bar(x + w/2, DSUN_AU,  w,
       label=r'DSUN\_OBS (ephemeris)')
for j in range(3):
    diff_pct = 100 * (D_DER_AU[j] - DSUN_AU[j]) / DSUN_AU[j]
    ax.annotate(f'{diff_pct:+.2f}%',
                xy=(x[j], max(D_DER_AU[j], DSUN_AU[j]) + 0.003),
                ha='center', va='bottom')
ax.set_xticks(x)
ax.set_xticklabels(DATES_SHORT)
ax.set_ylabel('Sun-observer distance (AU)')
ax.set_xlabel('Observation date')
ax.legend()
ax.set_ylim(0.985, 1.045)
fig.tight_layout()
fig.savefig(f"{OUT_DIR}/fig7_distance_comparison.pdf",
            dpi=150, bbox_inches='tight')
plt.close()
```

# References

[1] I. Kåsa, "A circle fitting procedure and its error analysis," *IEEE Trans. Instrum. Meas.*, **25**, 8–14, 1976.

[2] A. Prša et al., "Nominal values for selected solar and planetary quantities: IAU 2015 Resolution B3," *Astron. J.*, **152**, 41, 2016.

[3] D. Tripathi et al., "The Solar Ultraviolet Imaging Telescope on board Aditya-L1," *Sol. Phys.*, **300**, 30, 2025. arXiv:2501.02274, https://arxiv.org/abs/2501.02274.

[4] G. Taubin, "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, **13**(11), 1115–1138, 1991.

[5] W. T. Thompson, "Coordinate systems for solar image data," *Astron. Astrophys.*, **449**, 791–803, 2006.

[6] T. M. Brown and J. Christensen-Dalsgaard, "Accurate determination of the solar photospheric radius," *Astrophys. J.*, **500**, L195–L198, 1998.

[7] M. Haberreiter, W. Schmutz, and A. G. Kosovichev, "Solving the discrepancy between the seismic and photospheric solar radius," *Astrophys. J.*, **675**, L53–L56, 2008.

[8] M. Meftah et al., "Solar radius determined from PICARD/SODISM observations and extremely weak wavelength dependence in the visible and the near-infrared," *Astron. Astrophys.*, **616**, A64, 2018.