Modelling the Annual Cycle of Daily Antarctic Sea Ice Extent, 1979-2023

SARAH VILLHAUER,[1*] MARILYN RAPHAEL,[1], MARK HANDCOCK,[1]

[1] *University of California, Los Angeles*
[*] *svillhauer@g.ucla.edu*

1. Forenote

We divide our explanations of the Traditional Annual Cycle and Amplitude-Phase Adjusted Annual Cycle (APAC) into three subsections:

- *What is the ... ?* Provides the definition of each cycle, as presented by Handcock and Raphael (2020).
- *Numerical methods* Explains how each cycle was numerically computed.
- *Figures* Provides figures of the cycles for years in which the winter maximum was below the inter-decile range (1982, 1986, 1992, 2002, 2008, 2017, 2018, 2022) and for years in which the summer minimum was below the inter-decile range (1984, 1993, 1997, 2017, 2018, 2022).

2. Data

While daily sea ice extent data is available from the end of 1978 to present day (2024), we calculate the Traditional Annual Cycle using data spanning 1979-2023 in order to maintain consistency across calculations of different cycles. Calculating the Amplitude-Phase Adjusted Annual Cycle (APAC) for any designated year requires knowing the maximum extent in the following year; 2024 data does not yet include the annual maximum, thus we disregard 2024 data. Sea ice extent data was not interpolated, as our methods do not require complete temporal data record.

3. Calculating the Traditional Annual Cycle

3.1. What is the Traditional Annual Cycle?

Decomposition of Antarctic sea ice extent with the Traditional Annual Cycle is represented as:

$$\text{extent(t)} = a_T(\text{doy(t)}) + \alpha(\text{t}) \tag{1}$$

where $\text{t} = T_0, \ldots, T$.

- extent(t) is the recorded (satellite-observed) sea ice extent occurring on t, where t is time expressed as a decimal years (e.g. 1981.00958 is equivalent to January 3, 1981).
- doy(t) converts time expressed as decimal years to day of the year (which we define from 0 to 364) (e.g. 1981.00958 is equivalent to day 2).
- a_T is the annual shape function, with $a_T(\text{s})$ representing the annual cycle shape value for day of the year, s.
- $\alpha(t)$ is the anomaly between recorded sea ice extent and the annual shape function $a_T(\text{s})$ at time t.

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

- e.g. t = 1981.00958 is equivalent to Julian day 3, or s = 3. Recorded continental Antarctic sea ice extent at t = 1981.00958 is $5.84701 \times 10^6\,\mathrm{km}^2$, while $a_T(2) = 6.77012 \times 10^6\,\mathrm{km}^2$. $\alpha(\mathrm{t}) = \mathrm{extent(t)} - a_T(\mathrm{doy(t)}) \rightarrow \alpha(1981.00958) = \mathrm{extent}(1981.00958) - a_T(\mathrm{doy}(1981.00958)) \rightarrow \alpha(1981.00958) = \mathrm{extent}(1981.00958) - a_T(2) \rightarrow \alpha(1981.00958) = 5.84701 - 6.77012 = -0.9231 \times 10^6\,\mathrm{km}^2$

- In this paper, $T_0 = 1979.00136$ and $T_{\mathrm{end}} = 2023.99863$.

The annual cycle is traditionally estimated by $a_T(s)$:

$$a_T(s) = \frac{1}{\sum_{\mathrm{t:doy(t)=s}} 1} \sum_{\mathrm{t:doy(t)=s}} \mathrm{extent(t)} \tag{2}$$

where $\sum_{\mathrm{t:doy(t)=s}} 1 = 45$ is the number of years of data, where $\mathrm{t} = T_0, \ldots, T = 1979.00136, \ldots, 2023.99863$, and where s = 0, .., 364 (day of the year).

NOTE: Therefore, the Traditional Annual Cycle $a_T(\mathrm{s})$ is simply the daily mean of sea ice extent.

3.2. Numerical Methods

Using total continental Antarctic sea ice extent, for example.

3.2.1. Step One: Compute the daily mean sea ice extent

$$a_T(\mathrm{s}) = \begin{bmatrix} a_{T_0} \\ a_{T_1} \\ \vdots \\ a_{T_{364}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sum_{\mathrm{t:doy(t)=0}} 1} \sum_{\mathrm{t:doy(t)=0}} \mathrm{extent(t)} \\ \frac{1}{\sum_{\mathrm{t:doy(t)=1}} 1} \sum_{\mathrm{t:doy(t)=1}} \mathrm{extent(t)} \\ \vdots \\ \frac{1}{\sum_{\mathrm{t:doy(t)=364}} 1} \sum_{\mathrm{t:doy(t)=364}} \mathrm{extent(t)} \end{bmatrix} = \begin{bmatrix} \frac{1}{45}(\mathrm{extent}(1979.001369) + \ldots + \mathrm{extent}(2023.001369)) \\ \frac{1}{45}(\mathrm{extent}(1979.004109) + \ldots + \mathrm{extent}(2023.004109)) \\ \vdots \\ \frac{1}{45}(\mathrm{extent}(1979.998630) + \ldots + \mathrm{extent}(2023.998630)) \end{bmatrix}$$

$$= \begin{bmatrix} 7.126341 \\ 6.945984 \\ \vdots \\ 7.279182 \end{bmatrix}$$

3.3. Figures

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

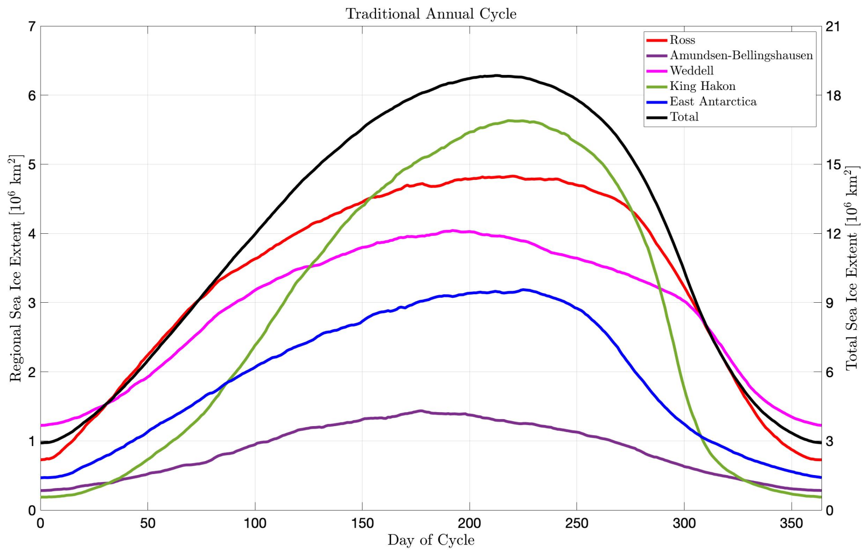

Fig. 1. The Traditional Annual Cycle a_T(s) of Antarctic sea ice extent, or the daily mean of sea ice extent from 1979-2023. Plotted such that day 0 of the cycle is when sea ice extent is at its annual minimum.

4. Calculating the Invariant Annual Cycle

4.1. What is the Invariant Annual Cycle?

The Invariant Annual Cycle a_I(s) models a_T(s) as a composition of cubic parabolas which join at their corresponding endpoints such that a'_I (first derivative), a''_I (second derivative), and a'''_I (third derivative) are all continuous. Balance between accuracy of fit to a_T(s) and smootheness is obtained through choosing a_I(s) to minimize penalized square error (PSE):

$$\mathrm{PSE}_\lambda(a) = \sum_{t=T_0}^{T} (\mathrm{extent(t)} - a_I(\mathrm{doy(t)}))^2 + \lambda \int_0^{365} a''_I(\mathrm{s})^2 \mathrm{ds} \quad \lambda > 0 \tag{3}$$

- extent(t) is the recorded (satellite-observed) sea ice extent occurring on t, where t is time expressed as a decimal years (e.g. 1981.00958 is equivalent to January 3, 1981).
- a_I is the invariant shape function, with a_I(s representing the invariant cycle shape value for day of the year, s.
- a''_I(s) is the second derivative of a_I(s).
- λ is the smoothing parameter.As $\lambda \to 0$, the function becomes an interpolating spline and will overfit the data. As $\lambda \to \infty$, smoothness dominates over accuracy of fit.

4.2. Numerical Methods

Reinsch's (1967) and Craven and Wahba's (1979) method for fitting smoothing splines under the penalized least squares approach is implemented using the csapsGCV function in MATLAB, developed by Matthew Taliaferro.

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

4.2.1. **Step One: Define x and y vectors**

Define the x vector.
s (day of the year) is the x vector.

$$x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{364} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 364 \end{bmatrix}$$

Normalize the x vector.

$$x = \begin{bmatrix} \frac{x_1 - \min(x)}{\max(x) - \min(x)} \\ \frac{x_2 - \min(x)}{\max(x) - \min(x)} \\ \vdots \\ \frac{x_{365} - \min(x)}{\max(x) - \min(x)} \end{bmatrix} = \begin{bmatrix} \frac{0-0}{364-0} \\ \frac{1-0}{364-0} \\ \vdots \\ \frac{364-0}{364-0} \end{bmatrix} = \begin{bmatrix} 0 \\ 0.0027 \\ \vdots \\ 1 \end{bmatrix}$$

Define the y vector.
$a_T(\mathrm{s})$ is the y vector, using $a_T(\mathrm{s})$ for total Antarctic sea ice extent for example.

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_364 \end{bmatrix} = \begin{bmatrix} \text{extent}(T_0) \\ \text{extent}(T_2) \\ \vdots \\ \text{extent}(T_{364}) \end{bmatrix} = \begin{bmatrix} 7.12634 \\ 6.94598 \\ \vdots \\ 7.27918 \end{bmatrix}$$

4.2.2. **Step 2: Calculate the difference matrix Q**

The matrix Q is a tridiagonal matrix with $(n+1)$ rows and $(n-1)$ columns where:

- $q_{i-1,i} = \frac{1}{h_{i-1}}$
- $q_{i,i} = \frac{-1}{h_{i-1}} - \frac{1}{h_i}$
- $q_{i+1,i} = \frac{1}{h_i}$

for $i = 0, 1, \ldots, n-1$, where $h_i = x_{i+1} - x_i$ and $n = \text{length(x)} - 1 = 364$.
Difference Vector h:

$$h = \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{364} \end{bmatrix} = \begin{bmatrix} x_1 - x_0 \\ x_2 - x_1 \\ \vdots \\ x_{364} - x_{363} \end{bmatrix} = \begin{bmatrix} 0.0027 - 0 \\ 0.0055 - 0.0027 \\ \vdots \\ 1 - 0.9963 \end{bmatrix} = \begin{bmatrix} 0.0027 \\ 0.0027 \\ \vdots \\ 0.0027 \end{bmatrix}$$

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

Matrix Q:

Initialize Q as a zero matrix.

$$Q = \begin{bmatrix} Q_{1,1} & Q_{1,2} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & Q_{1,363} \\ Q_{2,1} & Q_{2,2} & Q_{2,3} & \cdot & \cdot & \cdot & \cdot & \cdot & Q_{2,363} \\ Q_{3,1} & Q_{3,2} & Q_{3,3} & Q_{3,4} & \cdot & \cdot & \cdot & \cdot & Q_{3,363} \\ Q_{4,1} & Q_{4,2} & Q_{4,3} & Q_{4,4} & Q_{4,5} & \cdot & \cdot & \cdot & Q_{4,363} \\ & & \cdot & \cdot & \cdot & & & & \\ & & & \cdot & \cdot & \cdot & & & \\ & & & & \cdot & \cdot & \cdot & & \\ Q_{362,1} & Q_{362,2} & \cdot & \cdot & Q_{362,359} & Q_{362,360} & Q_{362,361} & Q_{362,362} & Q_{362,363} \\ Q_{363,1} & Q_{363,2} & \cdot & \cdot & \cdot & Q_{363,360} & Q_{363,361} & Q_{363,362} & Q_{363,363} \\ Q_{364,1} & Q_{364,2} & \cdot & \cdot & \cdot & \cdot & Q_{364,361} & Q_{364,362} & Q_{364,363} \\ Q_{365,1} & Q_{365,2} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & Q_{365,363} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ & & \cdot & \cdot & \cdot & & & & \\ & & & \cdot & \cdot & \cdot & & & \\ & & & & \cdot & \cdot & \cdot & & \\ 0 & 0 & \cdot & \cdot & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 \\ 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \end{bmatrix}$$

Define the diagonal values such that $q_{i+1,i} = \frac{1}{h_i}$, $q_{i,i} = \frac{-1}{h_{i-1}} - \frac{1}{h_i}$, and $q_{i-1,i} = \frac{1}{h_{i-1}}$.

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

$$
Q = \begin{bmatrix}
364 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\
-728 & 364 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\
364 & -728 & 364 & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\
0 & 364 & -728 & 364 & 0 & \cdot & \cdot & \cdot & 0 \\
 & & \cdot & \cdot & \cdot & & & & \\
 & & & \cdot & \cdot & \cdot & & & \\
 & & & & \cdot & \cdot & \cdot & & \\
0 & 0 & \cdot & \cdot & 0 & 364 & -728 & 364 & 0 \\
0 & 0 & \cdot & \cdot & \cdot & 0 & 364 & -728 & 364 \\
0 & 0 & \cdot & \cdot & \cdot & \cdot & 0 & 364 & -728 \\
0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0
\end{bmatrix}
$$

4.2.3. **Step 3: Calculate matrix** T

Matrix T is a positive definite tridiagonal matrix of order $n - 1$ where:

$$t_{i,i} = \frac{2(h_{i-1} + h_i)}{3}$$

$$t_{i,i+1} = t_{i+1,i} = \frac{h_i}{3}$$

Initialize T as a zero matrix.

$$
T = \begin{bmatrix}
t_{1,1} & t_{1,2} & \cdot & \cdot & \cdot & t_{1,363} \\
t_{2,1} & t_{2,2} & \cdot & \cdot & \cdot & t_{2,363} \\
 & & \cdot & & & \\
 & & & \cdot & & \\
 & & & & \cdot & \\
t_{363,1} & t_{363,2} & \cdot & \cdot & \cdot & t_{363,363}
\end{bmatrix}
=
\begin{bmatrix}
0 & 0 & \cdot & \cdot & \cdot & 0 \\
0 & 0 & \cdot & \cdot & \cdot & 0 \\
 & & \cdot & & & \\
 & & & \cdot & & \\
 & & & & \cdot & \\
0 & 0 & \cdot & \cdot & \cdot & 0
\end{bmatrix}
$$

Define the diagonal values such that $t_{i,i} = \frac{2(h_{i-1}+h_i)}{3}$

$$
T = \begin{bmatrix}
0.0037 & t_{1,2} & \cdot & \cdot & \cdot & t_{1,n-1} \\
t_{2,1} & 0.0037 & \cdot & \cdot & \cdot & t_{2,n-1} \\
 & & \cdot & & & \\
 & & & \cdot & & \\
 & & & & \cdot & \\
0 & 0 & \cdot & \cdot & \cdot & 0.0037
\end{bmatrix}
$$

and such that $t_{i,i+1} = t_{i+1,i} = \frac{h_i}{3}$.

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

$$T = \begin{bmatrix} 0.0037 & 9.1575e-04 & \cdot & \cdot & \cdot & 0 \\ 9.1575e-04 & 0.0037 & 9.1575e-04 & \cdot & \cdot & 0 \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ 0 & 0 & \cdot & \cdot & 9.1575e-04 & 0.0037 \end{bmatrix}$$

4.2.4. Step 4: Calculate matrices A and B

$$B = Q^T W^2 Q + pT$$

$$A = I - \left(\frac{W^2 Q}{B}\right) Q^T$$

where p is the smoothing parameter, and I and W are $n \times n$ (364x364) identity matrices. W is technically the weight matrix, or a vector of the standard deviations associated with the observed data. In our case, we leave weight unspecified, or equal to one.

4.2.5. Step 5: Calculate the spline coefficients

The spline function is defined as:

$$f(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

where:

- $a = y - p^{-1} W^2 Q c$
- $b_i = \frac{a_{i+1} - a_i}{h_i} - c_i h_i - d_i h_i^2$ for $i = 0, \ldots, n-1$
- $c = B^{-1} p Q^T y$ with $c_0 = c_n = 0$
- $d_i = \frac{c_{i+1} - c_i}{3h_i}$ for $i = 0, \ldots, n-1$

4.2.6. Step 6: Perform Generalized Cross Validation

Generalized Cross Validation measures model fit while penalizing complexity by minimizing:

$$V = \frac{N \times \mathrm{RSS}}{\mathrm{tr}^2}$$

over p. Where N = length(x) is the sample size, RSS is the residual sum of squares, and tr is the estimated degrees of freedom.

4.3. Figures

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

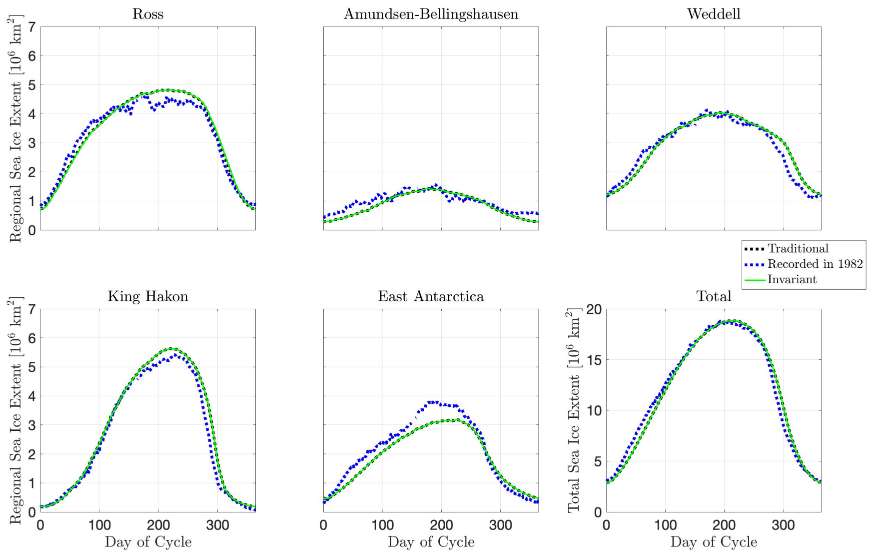

Fig. 2. The invariant cycle of Antarctic sea ice extent in 1982.

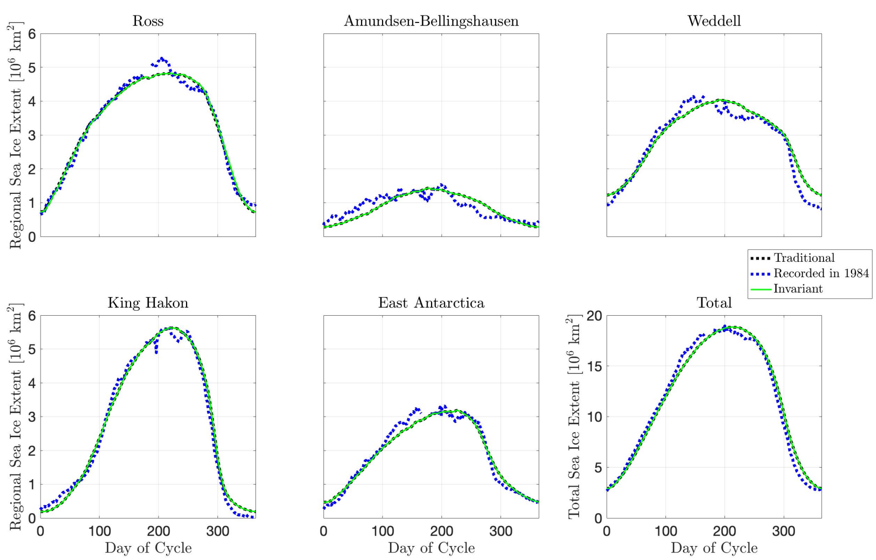

Fig. 3. The invariant cycle of Antarctic sea ice extent in 1984.

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

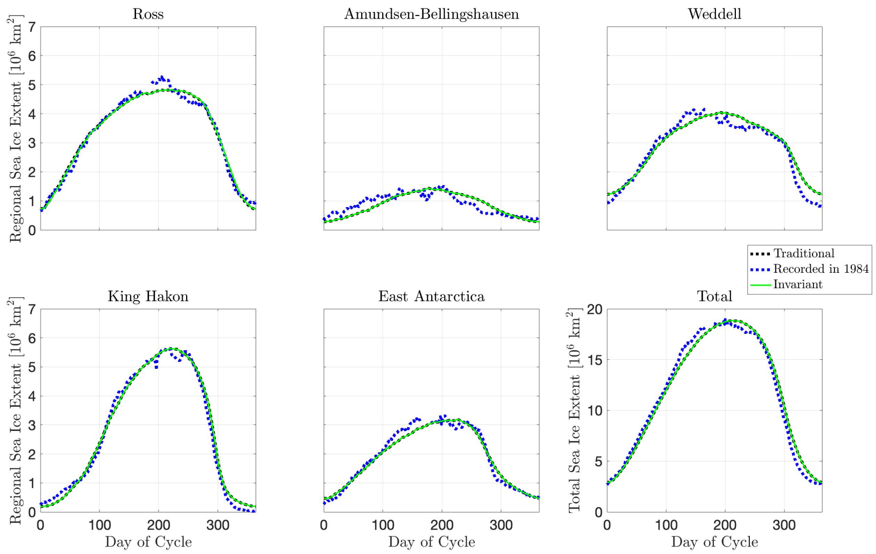

Fig. 4. The invariant cycle of Antarctic sea ice extent in 1985.

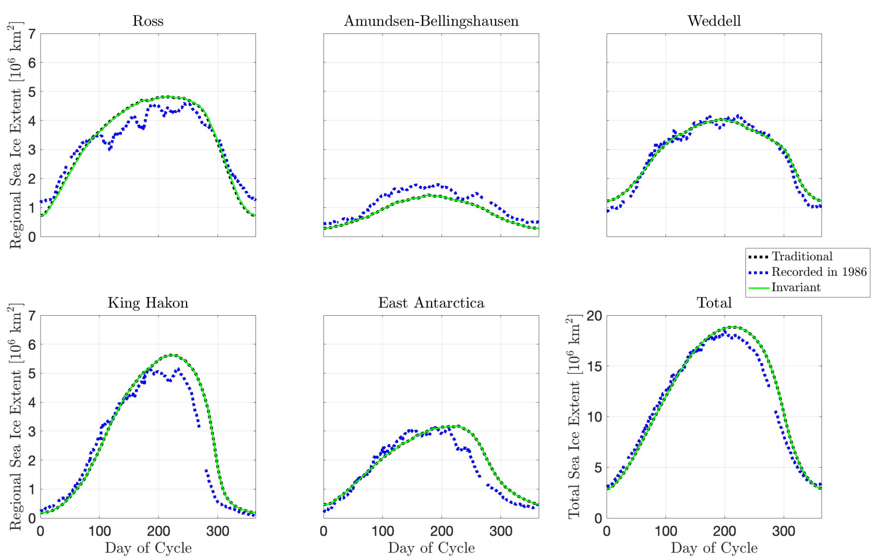

Fig. 5. The invariant cycle of Antarctic sea ice extent in 1986.

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

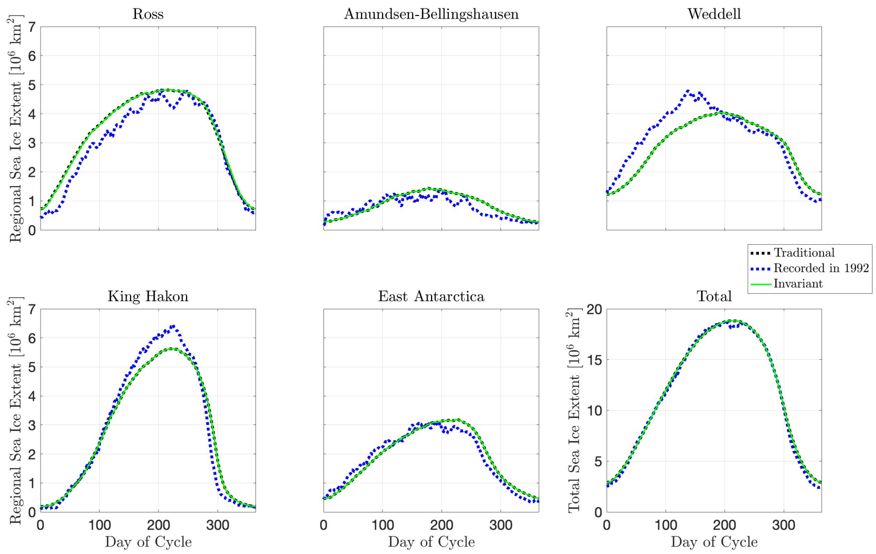

Fig. 6. The invariant cycle of Antarctic sea ice extent in 1992.

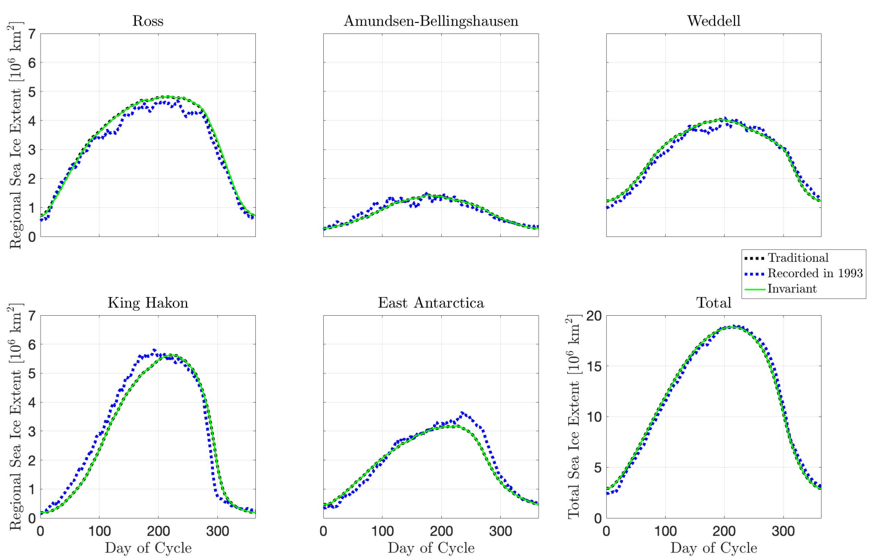

Fig. 7. The invariant cycle of Antarctic sea ice extent in 1993.

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

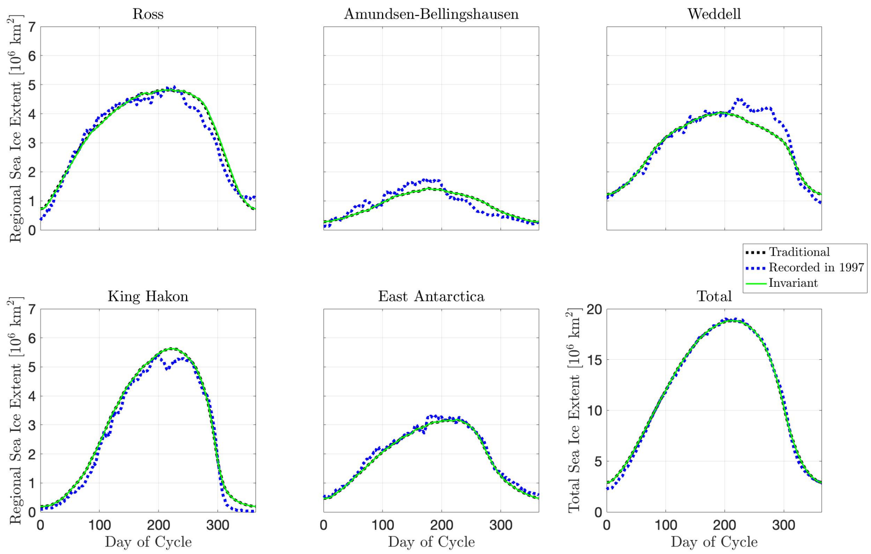

Fig. 8. The invariant cycle of Antarctic sea ice extent in 1997.

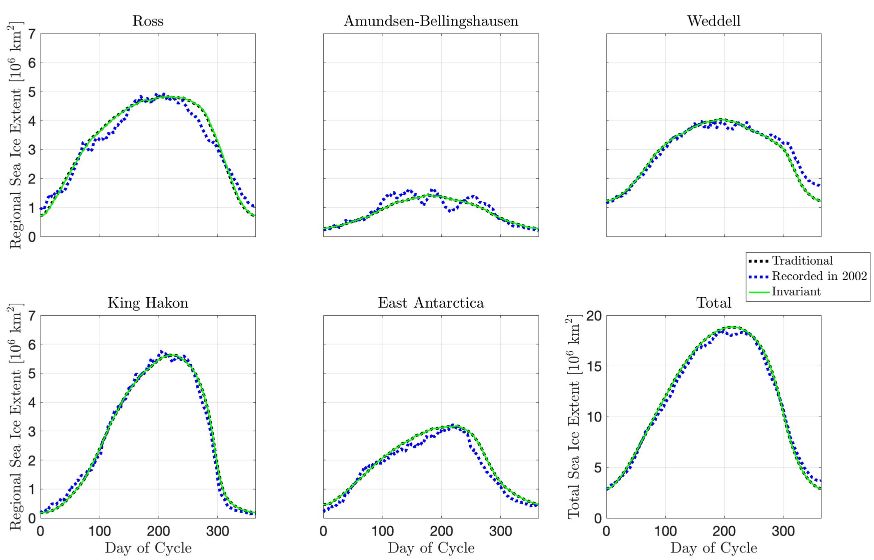

Fig. 9. The invariant cycle of Antarctic sea ice extent in 2002.

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

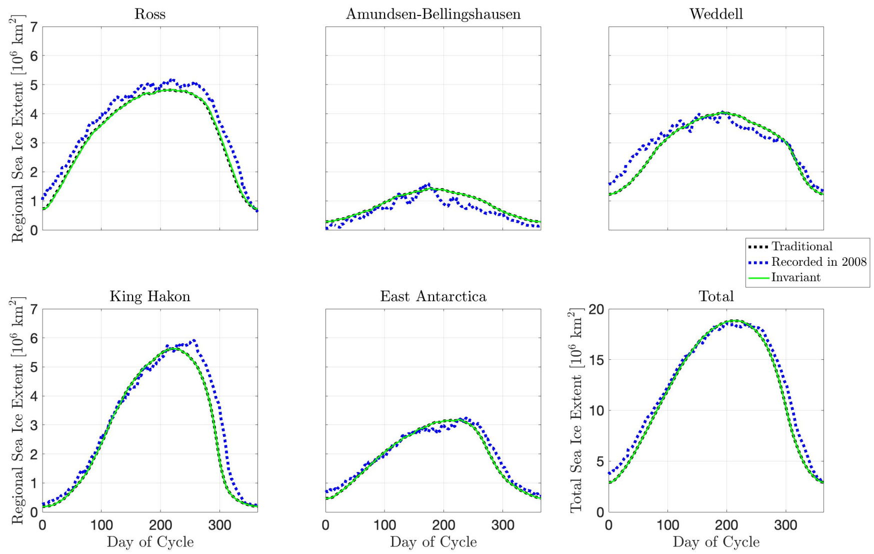

Fig. 10. The invariant cycle of Antarctic sea ice extent in 2008.

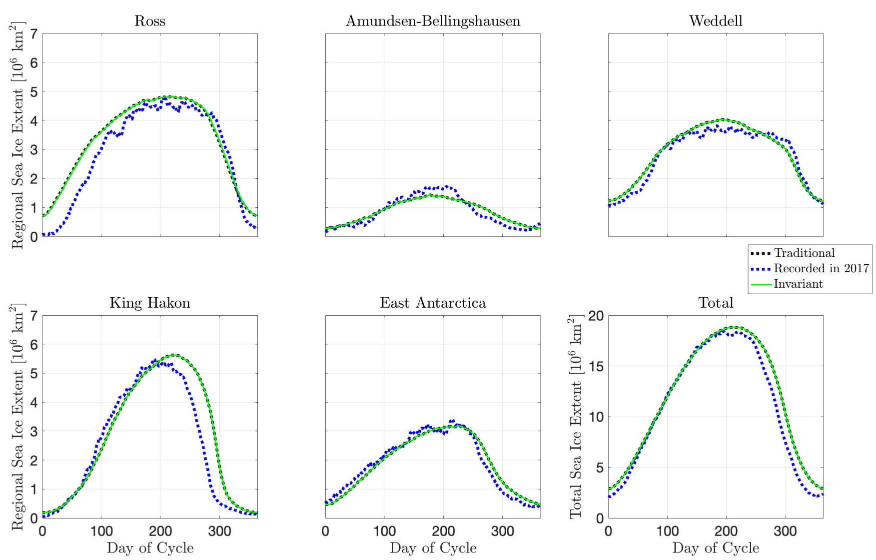

Fig. 11. The invariant cycle of Antarctic sea ice extent in 2017.

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

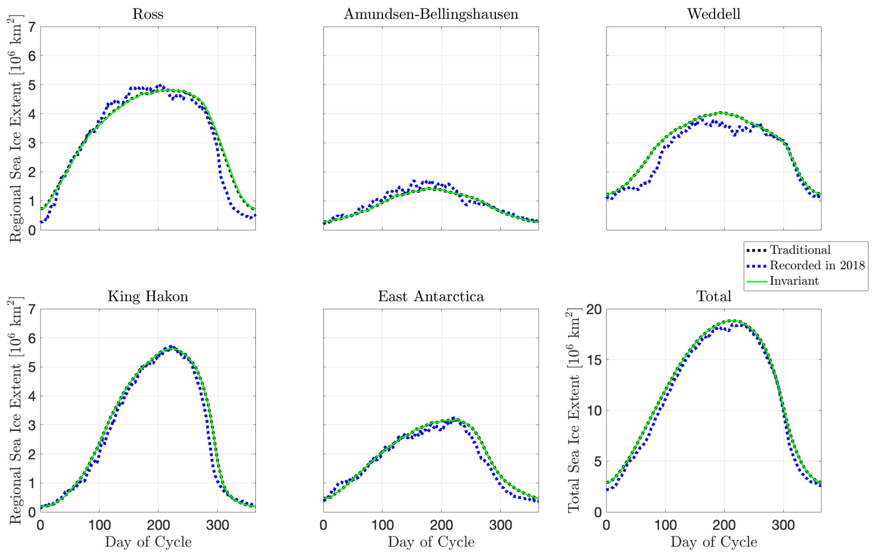

Fig. 12. The invariant cycle of Antarctic sea ice extent in 2018.

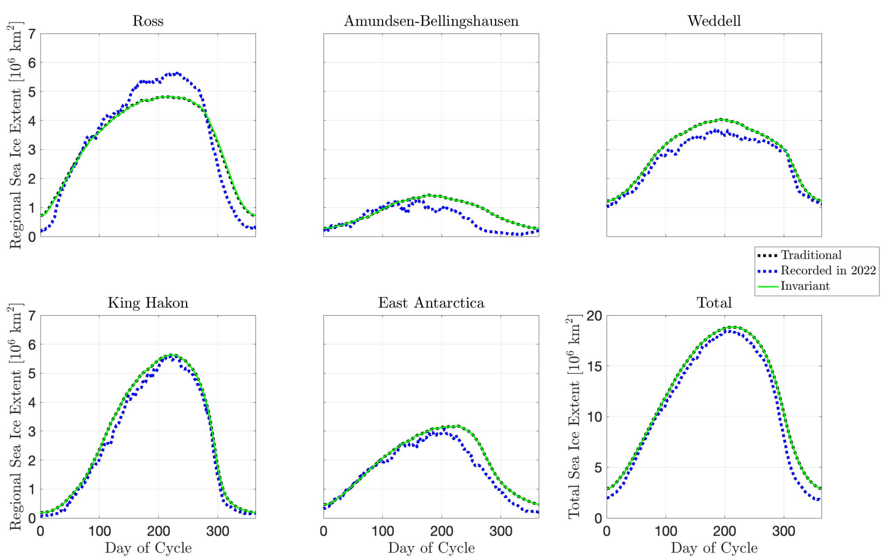

Fig. 13. The invariant cycle of Antarctic sea ice extent in 2022.

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

5. Calculating the Amplitude-Phase Adjusted Annual Cycle (APAC)

5.1. What is the Amplitude-Phase Adjusted Annual Cycle ?

The Amplitude-Phase Adjusted Annual Cycle (APAC) deforms the annual cycle in two ways: [1] by adjusting annual minimum and maximum extent, while maintaining the shape of the invariant cycle and [2] by adjusting the phase of the annual cycle, which influences both length and shape.

$$\text{extent(t)} = a_{APAC}(\text{phase(t)}, \text{min}\cdot\text{extent}(\text{year(t)}), \text{max}\cdot\text{extent}(\text{year(t)})) + \alpha(\text{t}) \tag{4}$$

where

$$a_{APAC} = u_A(\text{s})(\text{max} - \text{min}) + \text{min} \tag{5}$$

and

$$\text{phase(t)} = 365 \times \beta\left(\frac{\text{t} - \text{min}\cdot\text{extent}\cdot\text{day}(\text{year(t)})}{\text{max}\cdot\text{extent}\cdot\text{day}(\text{year(t)}+1) - \text{min}\cdot\text{extent}\cdot\text{day}(\text{year(t)})}; \beta(\text{year(t)})\right)$$
$$\text{min}\cdot\text{extent}\cdot\text{day}(\text{year}(t)) \leq t \leq \text{max}\cdot\text{extent}\cdot\text{day}(\text{year}(t)) \tag{6}$$

$u_A(\text{s})$ is chosen using Generalized Cross Validation, or to minimize penalized square error (PSE) over λ_{APAC} and $\{\beta_1(y) > 0, \beta_2(y) > 0\}_{1979}^{2022}$:

$$\text{PSE}_\lambda(a) = \sum_{t=T_0}^{T}\left(\frac{\text{t} - \text{min}\cdot\text{extent}(\text{year(t)})}{\text{max}\cdot\text{extent}(\text{year(t)}) - \text{min}\cdot\text{extent}(\text{year(t)})} - u_A(\text{phase}(\text{t}; \beta(\text{year(t)})))\right)^2 + \lambda_{APAC}\int_0^{365} u_A''(\text{s})^2 ds \quad \lambda_{APAC} > 0 \tag{7}$$

- extent(t) is the recorded (satellite-observed) sea ice extent occurring on t, where t is time expressed as decimal years (e.g., 1981.00958 is equivalent to January 3, 1981).
- $a_{APAC}(s) = a_A[\text{phase(t)}, \text{min}(\text{extent}(\text{year(t)})), \text{max}(\text{extent}(\text{year(t)}))]$ is the Amplitude-Phase Adjusted Annual Cycle.
- $\text{min}\cdot\text{extent}(\text{year}(t))$ is the annual minimum of sea ice extent.
- $\text{max}\cdot\text{extent}(\text{year}(t))$ is the annual maximum of sea ice extent.
- $u_A(\text{s})$ is the invariant annual cycle, computed from amplitude-adjusted data.
- $u_A''(\text{s})$ is the second derivative of $u_A(\text{s})$.
- phase(t) is the phase-adjusted day of the year for time t.
- β is the cumulative distribution function of a random variable parameterized by $\{\beta_1(y) > 0, \beta_2(y) > 0\}_{1979}^{2022}$.
- $\text{min}\cdot\text{extent}\cdot\text{day}(\text{year}(t))$ is the day of the year at which sea ice extent reaches its annual minimum (e.g. day 37 in 2016).
- $\text{max}\cdot\text{extent}\cdot\text{day}(\text{year}(t))$ is the day of the year at which sea ice extent reaches its annual maximum (e.g. day 255 in 2016).
- $\alpha(\text{t})$ is the anomaly between recorded sea ice extent and the annual shape function $a(\text{s})$ at time t.

5.2. Numerical Methods

Using 2016 total continental sea ice extent data for example.

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

5.2.1. Step 1: Scale data by minimum and maximum values for each year

In Section 4.2, we fit the invariant cycle to the traditional annual cycle $a_T(\mathrm{s})$, or the daily mean sea ice extent. To calculate $u_A(\mathrm{s})$, first we most deform the raw data to account for varying minimums and maximums every year.

Say we have a data matrix with (365) rows and (44) columns, with each column representing the years from 1979 to 2022.

$$\text{observed data} = \begin{bmatrix} \text{data}_{1,1979} & \cdot\ \cdot\ \cdot & \text{data}_{1,2022} \\ \text{data}_{2,1979} & \cdot\ \cdot\ \cdot & \text{data}_{2,2022} \\ & \vdots & \\ \text{data}_{365,1979} & \cdot\ \cdot\ \cdot & \text{data}_{365,2022} \end{bmatrix} = \begin{bmatrix} 7.3845 & \cdot\ \cdot\ \cdot & 5.8967 \\ 7.2729 & \cdot\ \cdot\ \cdot & 4.7376 \\ & \vdots & \\ 6.2944 & \cdot\ \cdot\ \cdot & 6.7420 \end{bmatrix}$$

$$\text{amp-adjusted data} = \begin{bmatrix} \frac{\text{data}_{1,1979} - \text{min}_{1979}}{\text{max}_{1979} - \text{min}_{1979}} & \cdot\ \cdot\ \cdot & \frac{\text{data}_{1,2022} - \text{min}_{2022}}{\text{max}_{2022} - \text{min}_{2022}} \\ \frac{\text{data}_{2,1979} - \text{min}_{1979}}{\text{max}_{1979} - \text{min}_{1979}} & \cdot\ \cdot\ \cdot & \frac{\text{data}_{2,2022} - \text{min}_{2022}}{\text{max}_{2022} - \text{min}_{2022}} \\ & \vdots & \\ \frac{\text{data}_{365,1979} - \text{min}_{1979}}{\text{max}_{1979} - \text{min}_{1979}} & \cdot\ \cdot\ \cdot & \frac{\text{data}_{365,2022} - \text{min}_{2022}}{\text{max}_{2022} - \text{min}_{2022}} \end{bmatrix}$$

$$= \begin{bmatrix} 0.4050 & \cdot\ \cdot\ \cdot & 0.2961 \\ 0.4054 & \cdot\ \cdot\ \cdot & 0.2877 \\ & \vdots & \\ 0.2035 & \cdot\ \cdot\ \cdot & 0.2944 \end{bmatrix}$$

5.2.2. Step 2: Calculate phase(t) for your designated year

Recall the definition of phase(t) is:

$$\text{phase}(\text{t}) = 365 \times \beta\left(\frac{\text{t} - \text{min}\cdot\text{extent}\cdot\text{day}(\text{year}(\text{t}))}{\text{max}\cdot\text{extent}\cdot\text{day}(\text{year}(\text{t})+1) - \text{min}\cdot\text{extent}\cdot\text{day}(\text{year}(\text{t}))}; \beta(\text{year}(\text{t}))\right) \quad \text{where} \quad \text{min}\cdot\text{extent}\cdot\text{day}(\text{year}(\text{t})) \le t \le \text{max}\cdot\text{extent}\cdot\text{day}(\text{year}(\text{t}))$$

Let

$$\text{xx} = \frac{\text{t} - \text{min}\cdot\text{extent}\cdot\text{day}(\text{year}(t))}{\text{max}\cdot\text{extent}\cdot\text{day}(\text{year}(t)+1) - \text{min}\cdot\text{extent}\cdot\text{day}(\text{year}(t))} \quad \text{where} \quad \text{min}\cdot\text{extent}\cdot\text{day}(\text{year}(t)) \le t \le \text{max}\cdot\text{extent}\cdot\text{day}(\text{year}(t))$$

Doing so results in a vector like:

$$\text{phase(t)} = 365 \times \beta(\text{xx}) = 365 \times \beta \begin{bmatrix} \text{xx}_0 \\ \text{xx}_1 \\ \vdots \\ \text{xx}_{37} \\ \text{xx}_{38} \\ \cdots \\ \text{xx}_{254} \\ \text{xx}_{255} \\ \vdots \\ \text{xx}_{364} \end{bmatrix} = 365 \times \beta \begin{bmatrix} \text{NaN} \\ \text{NaN} \\ \vdots \\ 0.0017 \\ 0.0034 \\ \vdots \\ 0.3733 \\ 0.3750 \\ \vdots \\ \text{NaN} \end{bmatrix}$$

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

Normalize the x vector.

$$x = \text{phase(t)} = 365 \times \beta(\text{xx}) = \begin{bmatrix} \text{NaN} \\ \text{NaN} \\ \vdots \\ 0.6250 \\ 1.2500 \\ \vdots \\ 136.2500 \\ 136.8750 \\ \vdots \\ \text{NaN} \end{bmatrix}$$

In this example, $\beta_1 = 0.9421$ and $\beta_2 = 0.9586$. xx is then linearly extrapolated in order to span 0 to 364.

Use the daily mean of amplitude-adjusted data as the y vector to calculate the invariant annual cycle $u_A(s)$, as described in Section 4.2, with the only difference being phase(t) is serving as our x vector. This means day 37 of the daily mean of the amplitude-adjusted data is used to compute day 0.6250 of the the Amplitude-Phase Adjusted Annual Cycle.

Step 3: Amplitude-adjust $u_A(s)$

$$u_A(s) = \begin{bmatrix} u_{A_0} \\ u_{A_1} \\ \vdots \\ u_{A_{581}} \\ u_{A_{582}} \end{bmatrix} = \begin{bmatrix} 0.2658 \\ 0.2589 \\ \vdots \\ 0.2832 \\ 0.2765 \end{bmatrix}$$

$$a_{APAC} = u_A(\text{s})(\max - \min) + \min = \begin{bmatrix} 0.2658 \\ 0.2589 \\ \vdots \\ 0.2832 \\ 0.2765 \end{bmatrix} (5.1716 - 0.3087) + 0.3087 = \begin{bmatrix} 6.9817 \\ 6.8696 \\ \vdots \\ 7.2635 \\ 7.1559 \end{bmatrix}$$

5.3. Figures

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

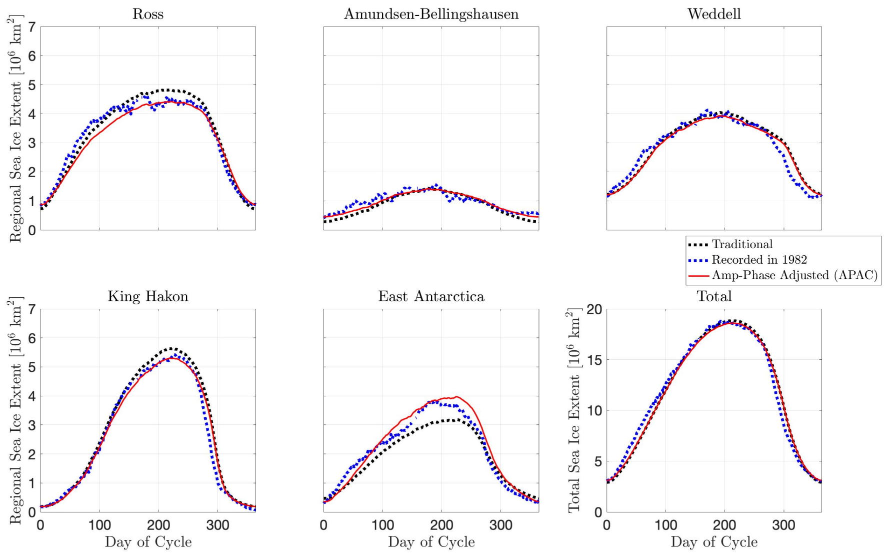

Fig. 14. Amplitude-Phase Adjusted cycle of Antarctic sea ice extent in 1982.

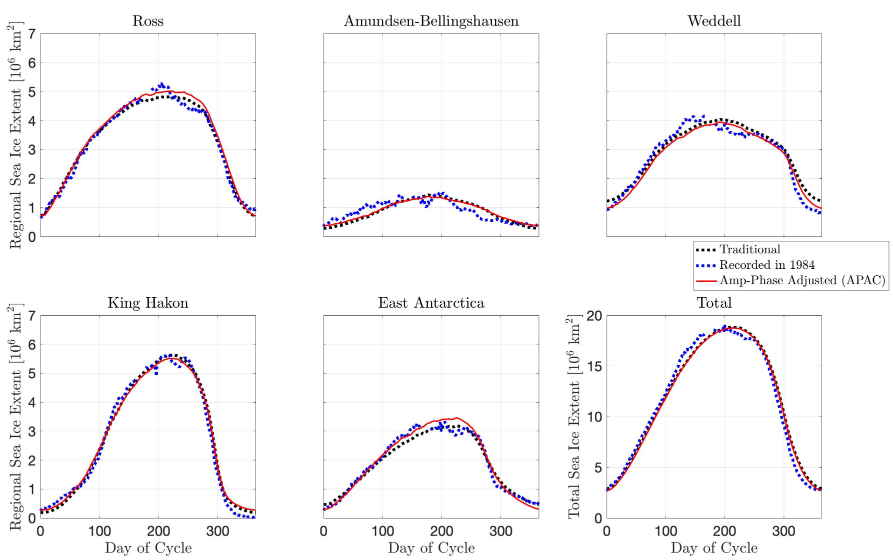

Fig. 15. Amplitude-Phase Adjusted cycle of Antarctic sea ice extent in 1984.

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

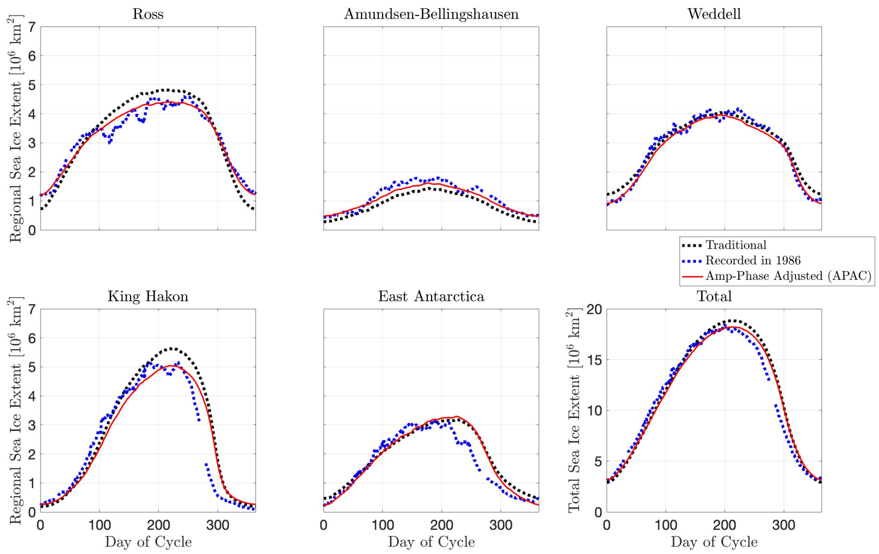

Fig. 16. Amplitude-Phase Adjusted cycle of Antarctic sea ice extent in 1986.

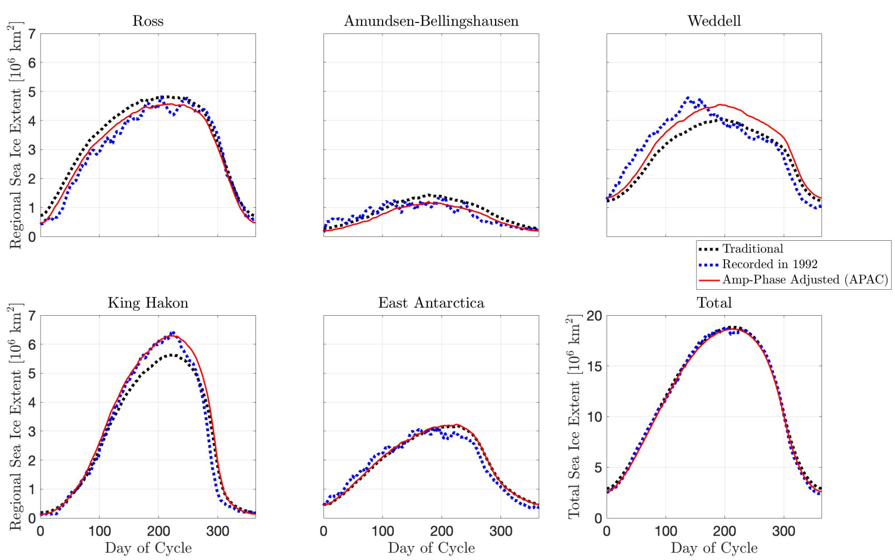

Fig. 17. Amplitude-Phase Adjusted cycle of Antarctic sea ice extent in 1992.

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

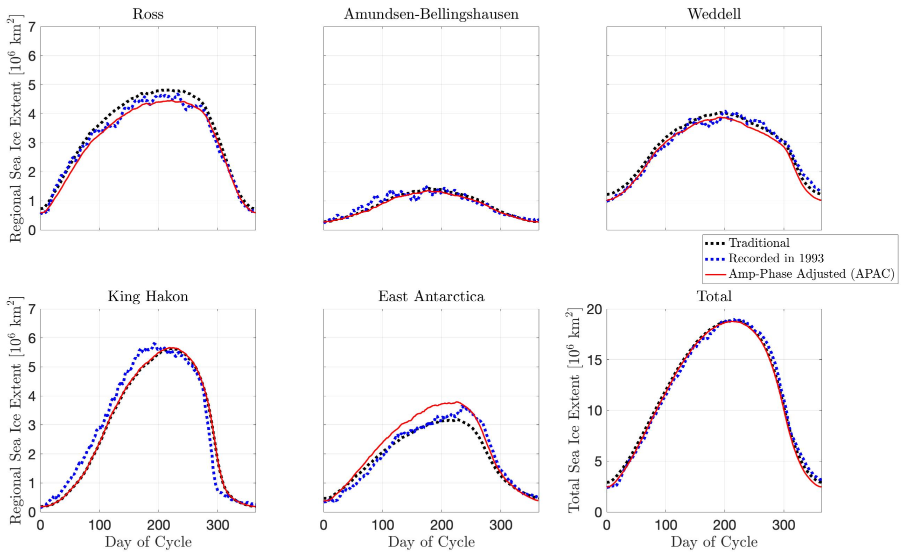

Fig. 18. Amplitude-Phase Adjusted cycle of Antarctic sea ice extent in 1993.

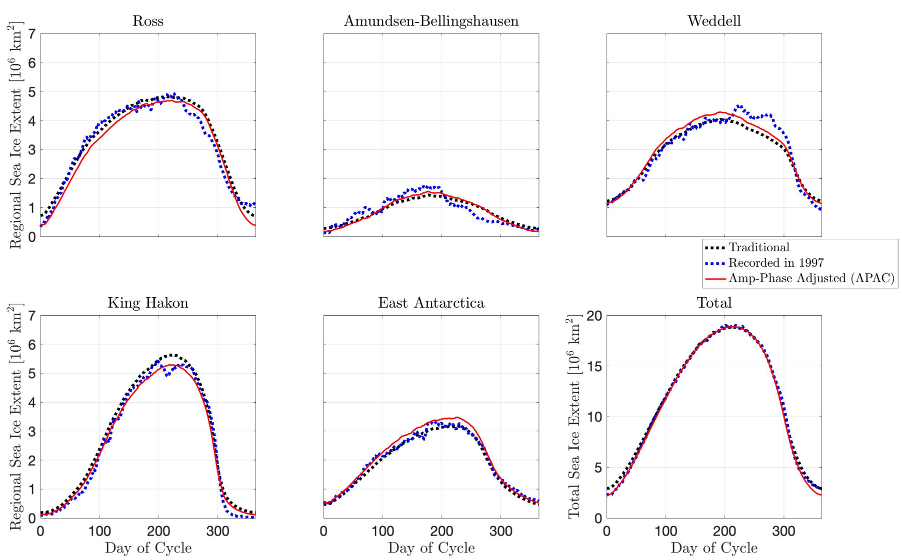

Fig. 19. Amplitude-Phase Adjusted cycle of Antarctic sea ice extent in 1997.

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

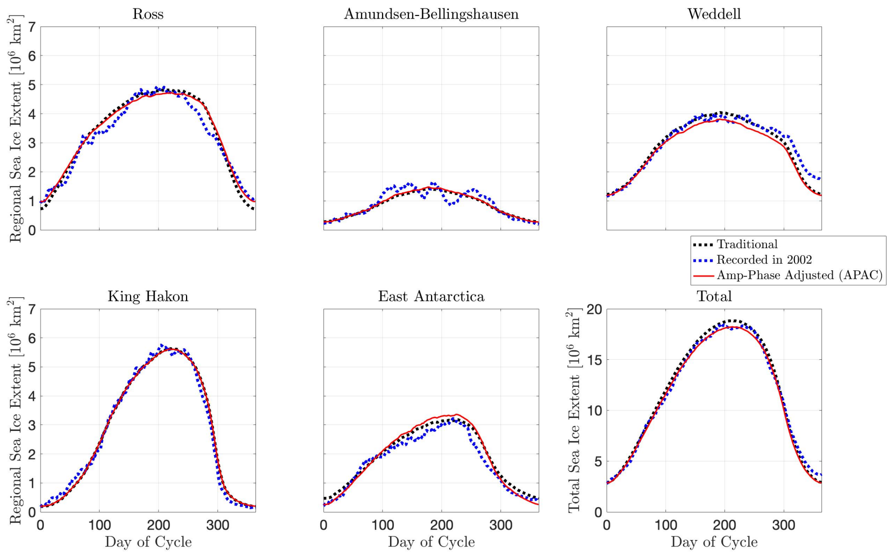

Fig. 20. Amplitude-Phase Adjusted cycle of Antarctic sea ice extent in 2002.

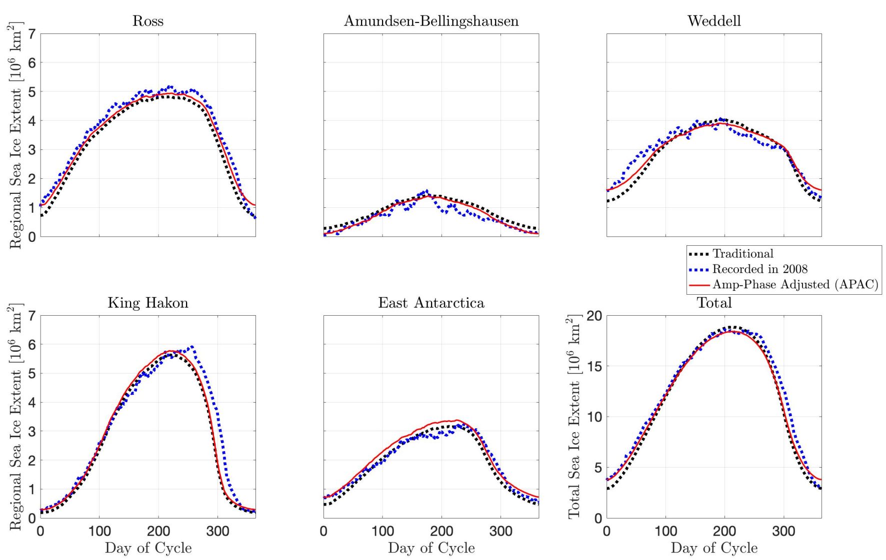

Fig. 21. Amplitude-Phase Adjusted cycle of Antarctic sea ice extent in 2008.

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

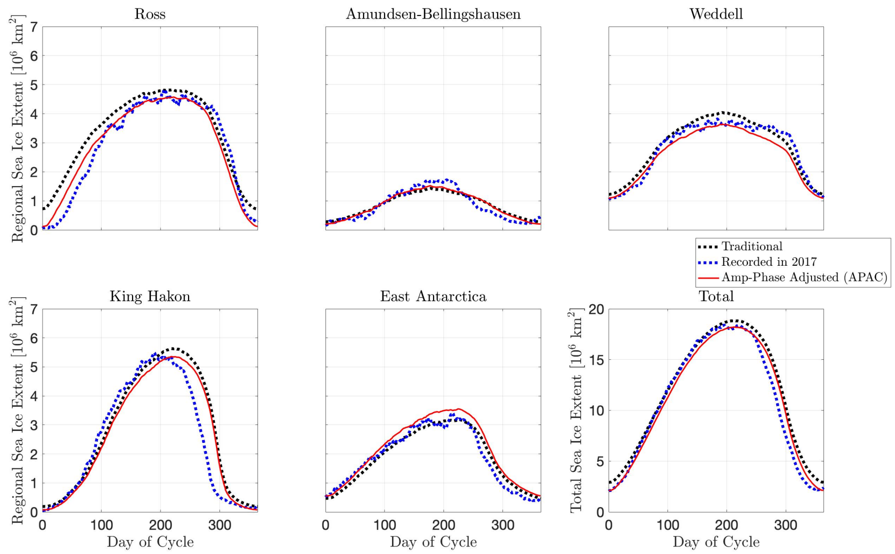

Fig. 22. Amplitude-Phase Adjusted cycle of Antarctic sea ice extent in 2017.

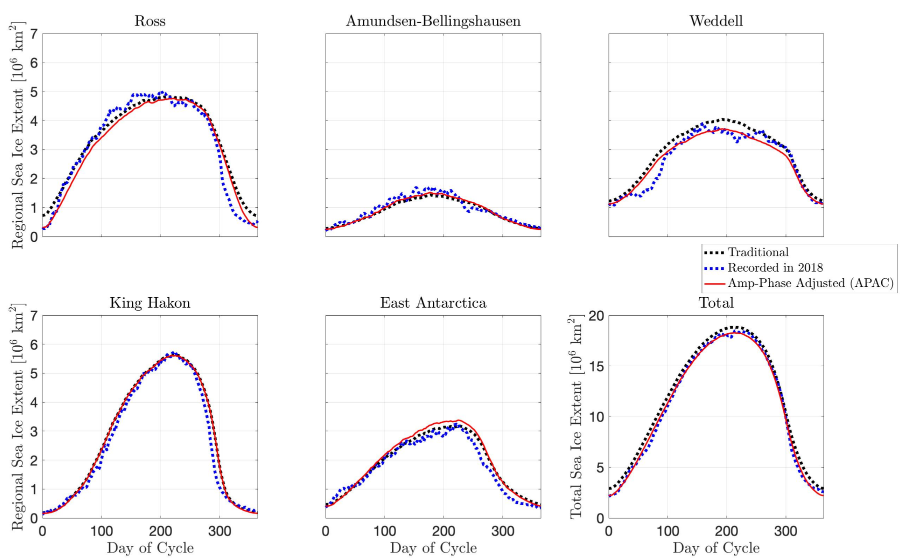

Fig. 23. Amplitude-Phase Adjusted cycle of Antarctic sea ice extent in 2018.

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.

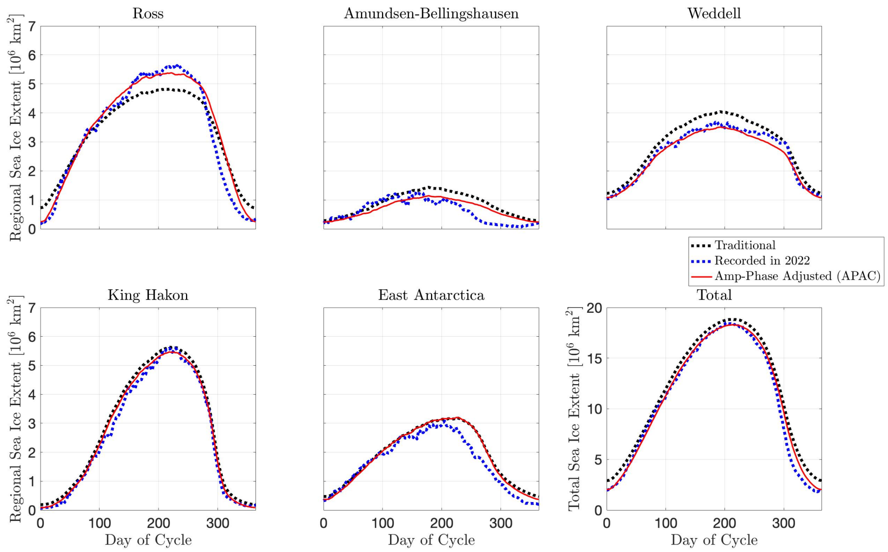

Fig. 24. Amplitude-Phase Adjusted cycle of Antarctic sea ice extent in 2022.

[1–3]

References

1. P. Craven and G. Whabba, “Smoothing noisy data with spline functions,” Numer. Math. **31** (1979).
2. M. S. Handcock and M. N. Raphael, “Modeling the annual cycle of daily antarctic sea ice extent,” The Cryosphere **14**, 2159–2172 (2020).
3. C. H. Reinsch, “Smoothing by spline functions,” Numer. Math. **10.3**, 177–183 (1967).

Data Availability: The data and MATLAB scripts that support the findings of this study are available on Github.