KANSAS GEOLOGICAL SURVEY OPEN-FILE REPORT 2024-6

Exploring drivers of groundwater recharge at Konza Prairie (Flint Hills region, Kansas, USA) using transfer function noise models

Authors: Ian Gambill[1,*], Sam Zipper[1,2,*], Matthew Kirk[3], Erin Seybold[1,2]

Affiliations:

1. Kansas Geological Survey, University of Kansas, Lawrence, Kansas
2. Department of Geology, University of Kansas, Lawrence, Kansas
3. Department of Geology, Kansas State University, Manhattan, Kansas

* Correspondence to ian.gambill@ku.edu and samzipper@ku.edu

Abstract: Intermittent streams, which lack continuous surface water flow throughout the year, constitute a large proportion of global waterways and influence downstream water quality and quantity. Stream intermittency has been increasing in the Great Plains over the past 40 years, and projected increased duration and severity of drought may cause more frequent or more severe intermittency in the future. Groundwater can be an important control over stream intermittency, and therefore it is important to understand groundwater-surface water interactions related to intermittent streamflow. We used climatological and hydrological data collected from the N04D watershed at the Konza Prairie Biological Station, a tributary to Kings Creek in northeastern Kansas, as inputs for transfer function noise models to assess potential drivers and their individual contribution to variations in groundwater levels in two groundwater monitoring wells, each drawing water from distinct geological units at Konza Prairie. Models for both wells were able to accurately reproduce historical groundwater levels (model Kling-Gupta Efficiency > 0.65). Results from the model for the upper Eiss Limestone Member indicate, in this unit, precipitation may be a joint driver of both stream levels and groundwater recharge. In the deeper Morrill Limestone Member, our models suggest that diffuse recharge and recharge via preferential pathways predominantly drive groundwater recharge to this unit. Our findings may guide future research on intermittent streams in Kansas and inform resource management.

Funding and Acknowledgments: This work was funded by National Science Foundation (NSF) awards OIA-2019603 for the Aquatic Intermittency Effects of Microbiomes in Streams (AIMS) project and DEB-2025849 for the Konza Prairie Long-Term Ecological Research (LTER) program. This is the final report for Ian Gambill as a 2023 participant in the Kansas Geological Survey (KGS) Geohydrology Internship Program. We appreciate helpful feedback and discussions with other AIMS, LTER, and KGS students, staff, and scientists.

Disclaimer: The Kansas Geological Survey made a conscientious effort to ensure the accuracy of this report. However, the Kansas Geological Survey does not guarantee this document to be completely free from errors or inaccuracies and disclaims any responsibility or liability for interpretations based on data used in the production of this document or decisions based thereon. This report is intended to make results of research available at the earliest possible date but is not intended to constitute formal publication.

1. Introduction

Intermittent streams are a type of non-perennial waterway and lack continuous surface water flow throughout the year (Messager et al., 2021). Non-perennial waterways constitute approximately half of the world's streams and rivers (Messager et al., 2021) and account for more than 60% of the total river length in the contiguous United States (Nadeau and Rains, 2007). Due to their ubiquity, non-perennial streams are referred to in various ways, including "intermittent," often denoting seasonal flow, and "ephemeral," which describes short-lived flows directly linked to precipitation (Busch et al., 2020). In this paper, we will use the term "intermittent" to refer to these streams, and "intermittency" will encompass streams exhibiting both intermittent and ephemeral flow characteristics. Intermittent streams perform several ecological services, including supporting diverse aquatic and riparian ecosystems and habitats (Datry et al., 2014), supporting water supply in downstream perennial streams or the aquifer through groundwater recharge (Shanafield et al., 2020; Shanafield and Cook, 2014), and influencing the transport of materials, nutrients, and biota within watersheds (Datry et al., 2018).

In the native tallgrass prairie of the Great Plains in the central United States, one of the most endangered biomes globally (Samson and Knopf, 1994), many stream systems are non-perennial flow regimes characterized by harsh intermittency and low predictability (Dodds et al., 2004; Samson and Knopf, 1994). Despite experiencing zero-flow conditions for significant portions of the year, these intermittent stream systems still exert an influence on downstream water quality (Dodds and Oakes, 2006; Dodds and Oakes, 2008). Even in areas where precipitation is increasing, it may be partially or fully counteracted by rising temperatures and/or the impacts of woody encroachment (Dodds et al., 2023; Keen et al., 2023; Sadayappan et al., 2023). Non-perennial streams are becoming drier across much of the United States, including the southern Great Plains, primarily due to increasing aridity (e.g., Hammond et al., 2021; Zipper, Hammond et al., 2021), and studies elsewhere have revealed that changes in stream intermittency are common in Europe (Tramblay et al., 2021) and Australia (Sauquet et al., 2021). Several studies have identified climatic aridity as an important control on stream intermittency (Hammond et al., 2021). However, other factors can be locally important, such as vegetation dynamics (Katz et al., 2012), subsurface hydrostratigraphic characteristics (Warix et al., 2021), groundwater pumping (Zipper, Farmer et al., 2022; Zipper, Popescu et al., 2022), and other anthropogenic modifications to the water cycle (Datry et al., 2022). Given the potential for increased drying and the spatial-temporal variability of intermittency in streams, it becomes imperative to understand the patterns and driving factors behind streamflow and stream intermittency dynamics in this region.

Characterizing the hydrological processes is challenging in intermittent stream systems. First, hydrologic records of these systems are frequently scarce, incomplete, and relatively short, which impose limitations on data availability (Krabbenhoft et al., 2022; Shook and Pomeroy, 2012). Second, there is a general lack of knowledge regarding the applicability of standard hydrologic indices, originally developed for perennial streams, to intermittent streams (Olden and Poff, 2003; Price et al., 2021). Because streamflow is particularly challenging to measure in the low-flow conditions commonly experienced in non-perennial streams (Seybold et al., 2023) and groundwater can be a locally important contributor to streamflow in many intermittent streams (Swenson et al., 2024; Warix et al., 2021; Zipper, Popescu et al., 2022), hydrological methods relying on groundwater levels from wells in non-perennial catchments may provide insight into the processes driving streamflow (Rinderer et al., 2019).

Groundwater recharge estimation remains a formidable challenge for hydrologists (Collenteur et al., 2021). To address this challenge, transfer function noise (TFN) models offer a relatively simple yet versatile approach applicable to various hydrological settings. TFN models synthesize one or more input time series (e.g., precipitation, potential evaporation, stream stage) to predict an output time series (e.g., groundwater levels). The early usage of TFN models to estimate recharge was a demanding task, necessitating a calibrated groundwater model to establish a relation between recharge flux and rainfall by using an additional hydrograph for the unsaturated zone (Besbes and De Marsily, 1984). However, recent advancements have led to the use of specific TFN models equipped with predefined response functions for analyzing groundwater levels (von Asmuth et al., 2002). This data-driven approach has been used to elucidate how groundwater levels respond to various drivers, such as precipitation, evaporation, and pumping, through the use of response functions. Although numerical groundwater flow models such as MODFLOW contain a more holistic representation of processes, TFN models offer several advantages over numerical models, including the ability to analyze numerous time series efficiently, reduced calibration parameter requirements, and the potential to inform the development of more complex numerical models (Bakker and Schaars, 2019). Additionally, the application of physically based response functions allows for the analysis of the individual contribution to the change in head from each distinct stress (e.g., Collenteur et al., 2019; von Asmuth et al., 2008, 2012), combining the interpretability and process information of numerical models with the computational simplicity of statistical approaches (Bakker and Schaars, 2019). This approach has demonstrated success in the analysis of groundwater dynamics (Collenteur et al., 2021), including in locations with intermittent streamflow regimes (Zipper, Popescu et al., 2022).

Here, our objective was to characterize groundwater level variation and inferred groundwater recharge using TFN models in the N04D watershed at the Konza Prairie Biological Station in the Flint Hills region of northeastern Kansas. We applied TFNs, driven by local meteorological data, to two distinct limestone aquifers at Konza Prairie. This approach was employed to gain insights into the factors potentially driving groundwater-surface water interactions and stream intermittency at the Konza Prairie and evaluate the potential utility of TFN models for sites with complex geology and stream-aquifer interactions.

2. Study Site and Methods

2.1. Konza Prairie

Konza Prairie, located in the Flint Hills region of northeastern Kansas (fig. 1), encompasses a complex and dynamic environment characterized by its distinctive geological and ecological features. The study specifically focuses on watershed N04D, a 1.2 km² headwater catchment within this area. This catchment, drained by an intermittent fourth-order stream, is predominantly covered by native prairie grasses, primarily warm-season grasses such as big bluestem, little bluestem, Indian grass, and switchgrass. However, Konza Prairie also has significant woody vegetation cover, including such species as burr oak, hackberry, and American elm in the riparian zone, and shrubs such as rough-leaf dogwood and smooth sumac on the hillslopes (e.g., Briggs et al., 2005; Sadayappan et al., 2023; Veach et al., 2015). The riparian corridors host various woody species, and there has been a notable increase in shrub cover in recent decades, particularly in areas with less frequent fire management (e.g., Sadayappan et al., 2023). The prairie experiences a midwestern continental climate with warm, wet summers and cold, dry winters. The mean annual precipitation is approximately 835 mm, with most rainfall occurring during the growing season from April to October (Hayden, 1998). The area has a mean annual temperature of about 11.7 °C (Nippert and Knapp, 2007). N04D has been subject to a regime of controlled burns, occurring every four years to study the impact of grassland burn frequency on woody plant abundance (Veach et al., 2015), and is grazed by native ungulates (American bison).

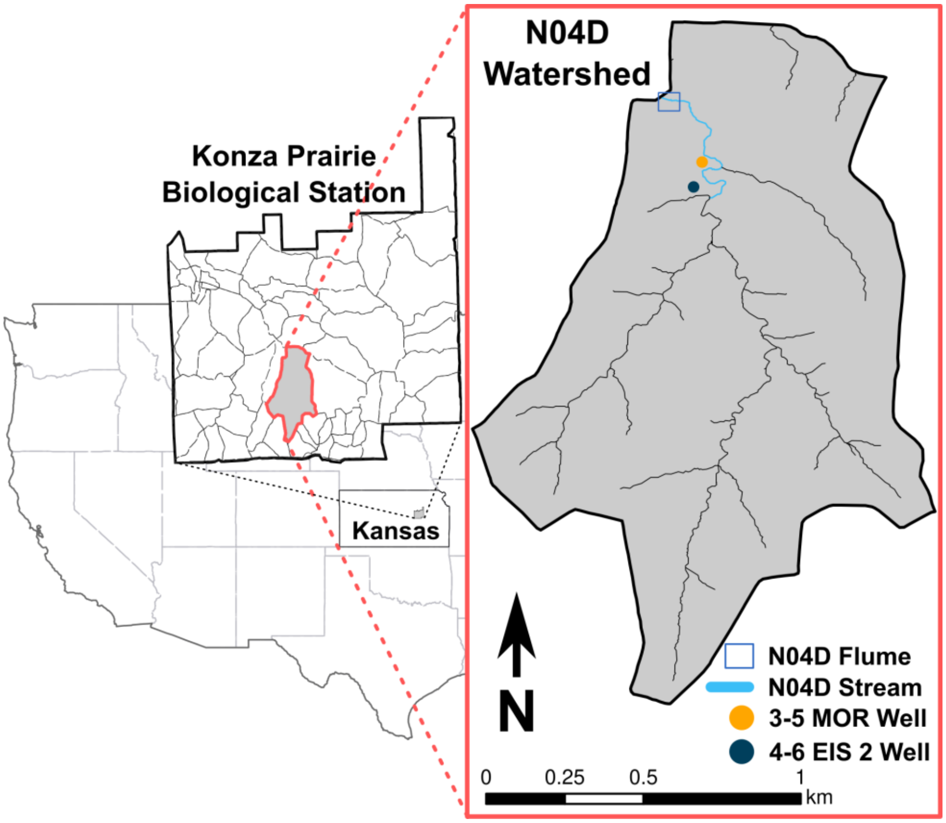


Figure 1. Watershed N04D within Konza Prairie Biological Station. Relevant monitoring wells and flume are located in the northern section of N04D. Modified from Hatley et al. (2023).

Geologically, Konza Prairie is situated on highly complex merokarst bedrock consisting of alternating layers of limestone and mudstone, with limestone units such as the Cottonwood, Morrill, and Eiss acting as the primary aquifers (fig. 2; Macpherson, 1996). Macpherson (1996) showed that within the studied geologic units, the lower Morrill exhibited a likely connection to the stream, while the uppermost unit (upper Eiss) showed a probable link to the stream, albeit with less certainty. These limestone aquifers, interspersed with low permeability mudstones, act as semi-isolated aquifers facilitating primarily horizontal groundwater flow (e.g., Hatley et al., 2023). Hydraulic conductivity within these units is highly variable, and groundwater flow is mostly accommodated by fractures and solution-enlarged pores (Hatley et al., 2023). Measured hydraulic conductivities of the Morrill and Eiss fall within typical values of other karstic settings

(Lewis et al., 2006); however, these values should not be generalized as accurate due to the aforementioned heterogeneity. Surface runoff contributes to streamflow during storms, but soil macropores and bedrock fractures predominantly route precipitation into the subsurface (e.g., Macpherson et al., 2008; Tsypin and Macpherson, 2012) and streamflow typically occurs when limestone aquifers fill to capacity and begin discharging to the surface (Costigan et al., 2015). As a result of these subsurface flow paths, water level data and geochemical signatures indicate that a significant portion of streamflow is sourced from groundwater (Hatley et al., 2023), with increasingly old groundwater as the stream network dries (Swenson et al., 2024). However, the area also experiences significant variability in water flow paths, both spatially and temporally, with evidence of cross-unit vertical transport in certain localized areas (e.g., Barry, 2018; Sullivan et al., 2020).

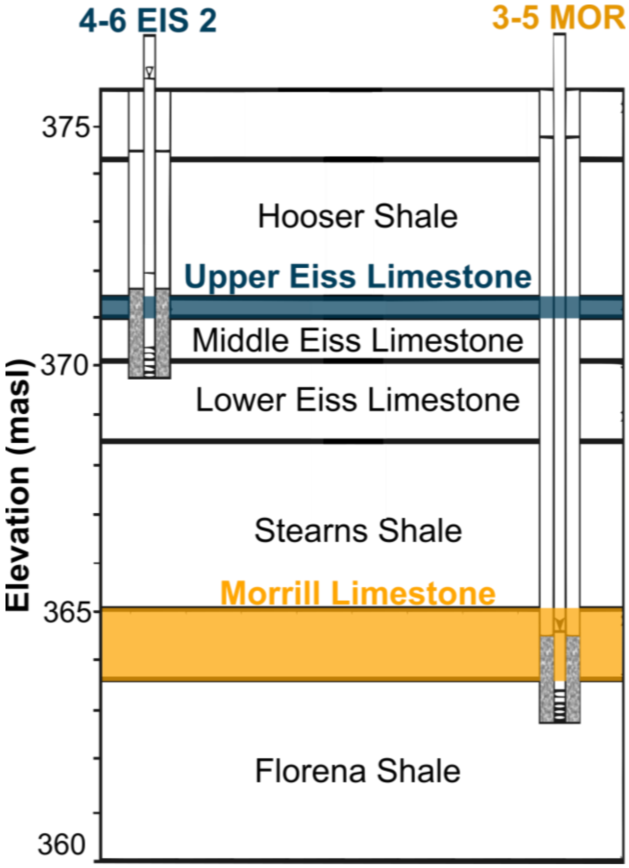


Figure 2. Diagram of major geological units in N04D watershed with 4-6 EIS 2 and 3-5 MOR groundwater monitoring wells. 4-6 EIS 2 draws from the geologically shallower upper Eiss (blue) and 3-5 MOR draws from the deeper Morrill (yellow). Note that the well screens and gravel packs (gray) overlap multiple units. Modified from Barry (2018).

2.2. Field Data

Data used in this study were collected at the Konza as part of the Konza Prairie Long-Term Ecological Research (LTER) program. All of these data and related information can be found at http://lter.konza.ksu.edu/data. Our study used data collected from two observation wells (Kirk and Macpherson, 2023) that measured groundwater levels in two distinct geological units, the upper Eiss and the Morrill (fig. 2). Groundwater level observations were measured at five-minute intervals using pressure transducers, which we aggregated into daily averages (fig. 3). Additionally, our study used several climatological data sets collected at Konza headquarters (fig. 4). These data included daily precipitation amounts (Nippert, 2023) collected at N04D flume — missing data were filled with measurements collected at the Konza headquarters; daily total short grass reference evapotranspiration (ET_0), which represents a surface completely covered with well-watered grass at a uniform height of 0.12 m, a surface resistance of 70 s m^{-1}, and an albedo of 0.23 (Moore, 2023); daily total climatic water balance (precipitation minus ET_0, as in Zipper, Popescu et al., 2022); and N04D stream stage (Dodds, 2023), collected at the flume shown in fig. 1 at five-minute intervals and aggregated into daily averages after we conducted outlier analysis. Here, we would like to acknowledge that short grass ET_0 is a standardized approach to estimate the atmospheric demand for water and is not intended to be an estimate of actual evapotranspiration in our system, which would require a characterization of plant growth stage, soil moisture and other stresses, and other factors affecting plant water use. These data, which will be referred to as stresses, were used as inputs for models that estimate recharge and simulate groundwater level.

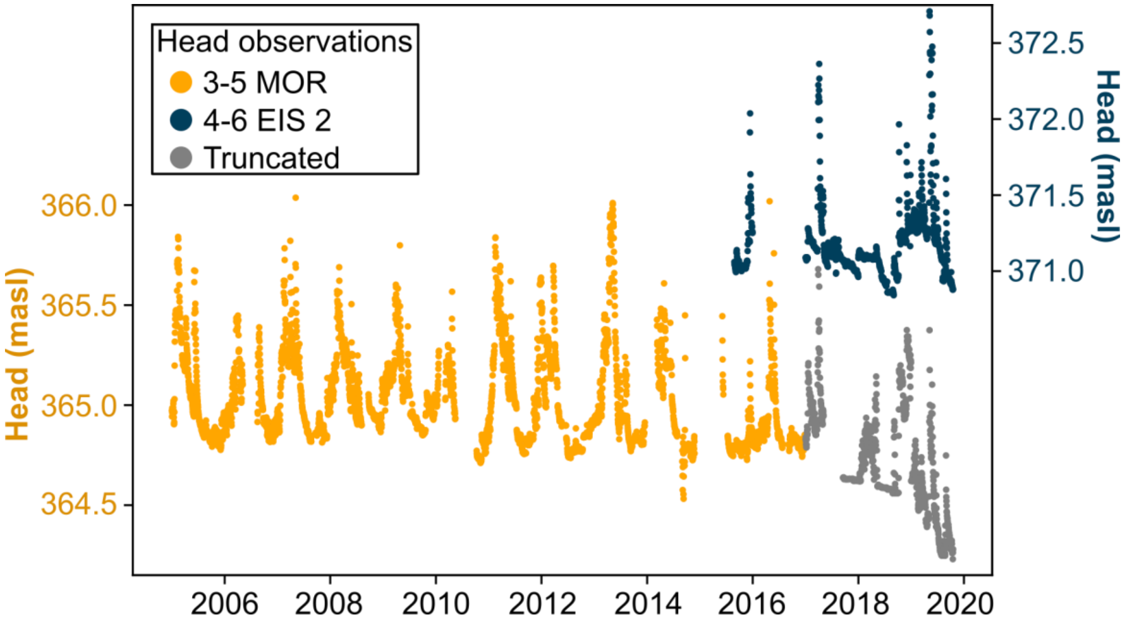


Figure 3. Head observations from 4-6 EIS 2 (blue) and 3-5 MOR (yellow) monitoring wells. Beginning in 2017, observation time series of 3-5 MOR was truncated due to a potentially erroneous downward trend. Truncated observations are shown in gray. Note the different y-axes.

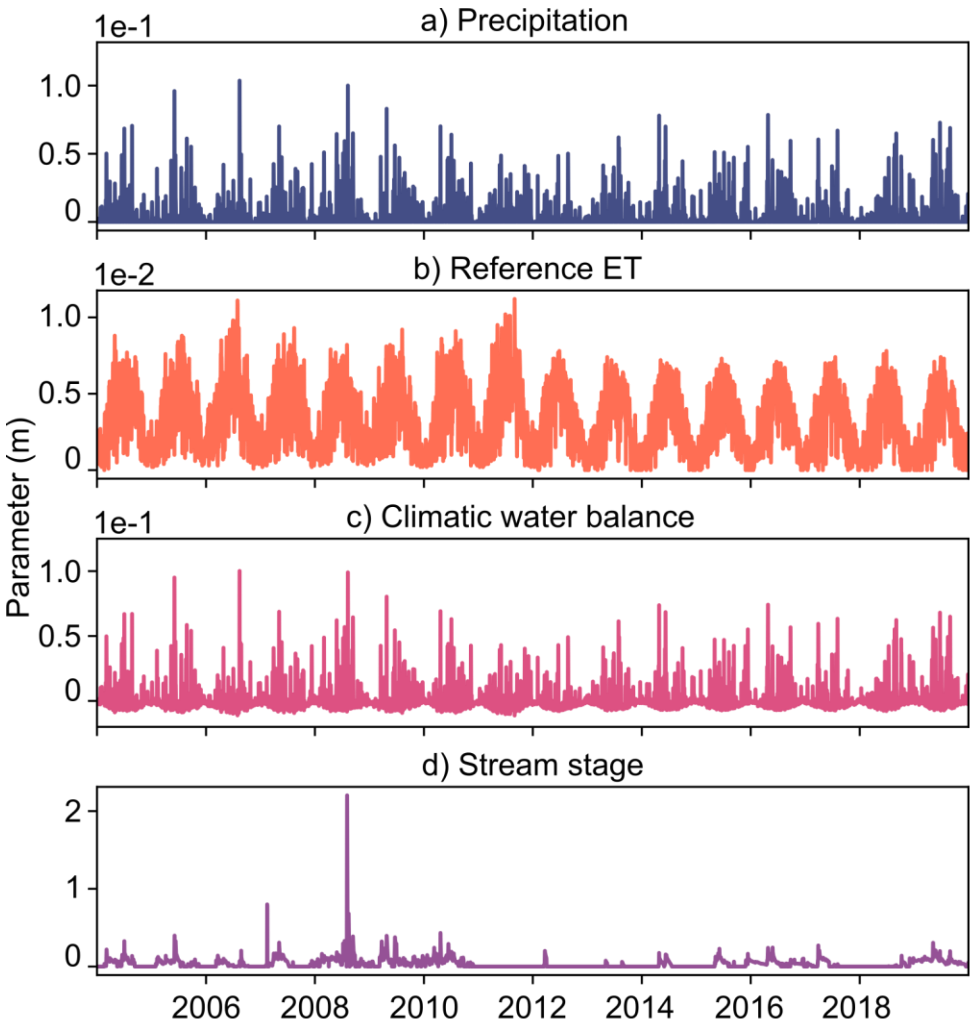


Figure 4. Climatological and hydrological data from Konza Prairie Biological Station used as stresses for TFN models.

2.3. Transfer function noise (TFN) models

TFN modeling attempts to explain an observed time series by synthesizing one or more stress time series into a simulated time series, fitting specific parameters associated with each stress to match observed water levels. Because only a point in space is modeled, represented by head fluctuations in a monitoring well, TFN models provide information about localized hydrological processes and may not be generalizable beyond the study well. However, TFN models require substantially less input data than process-based numerical models and often provide similar or better model fits (e.g., Bakker and Schaars, 2019). A basic TFN model is described by the following equations (e.g., Collenteur et al., 2019):

$$h(t) \ = \ \sum_{m\,=\,1}^{M} \quad h_m(t) \ + \ d \ + \ r(t), \tag{1}$$

where $h(t)$ is observed head, $h_m(t)$ is the contribution of stress m to the head, d is the base elevation of the model, and $r(t)$ isthe residual. Models can include an arbitrary number of stresses that may include precipitation, evaporation, stream stage, and groundwater extractions. The contribution to stress m is calculated through convolution:

$$h_m(t) \ = \ \int_{-\infty}^{1} \quad S_m(\tau)\theta_m(t-\tau)d\tau, \tag{2}$$

where S_m is the time series of stress m, and θ_m is the impulse response (IR) function for stress m. These models are used to estimate $h_r(t)$, the contribution of recharge to head fluctuations through convolution with $R(\tau)$, a recharge flux, with a predefined IR function $\theta(t)$ (e.g., von Asmuth et al., 2002):

$$h_r(t) \ = \ \int_{-\infty}^{1} \quad R(\tau)\theta(t-\tau)d\tau. \tag{3}$$

The IR functions in TFNs are lumped parameter models that translate hydrological drivers, such as precipitation, into changes in groundwater levels (von Asmuth et al., 2002). Because an IR function can take several different shapes, it is able to characterize head response to an impulse from a driver (e.g., a precipitation event) and can describe a relation between the impulse and the response (head) (e.g., Jemeļjanova et al., 2023). The following subsections describe some common IR functions and the processes they are often used to characterize. All the response functions are summarized in table 1, including the processes or responses that each IR is used to characterize (Collenteur et al., 2019; Collenteur et al., 2021).

Table 1. Summary of impulse response details.

Impulse Response	**Abbreviation**	**Simulated Process/Response of head**
Exponential	Exp	Rapid response to stress
Double Exponential	DblExp	Initial quick response that becomes slower, for multi-aquifer systems
Gamma	Gam	Response from areal recharge
One	—	Instant response with no lag
Four Parameter	FourP	Delayed response, can be used to capture compound recharge from the root zone, the unsaturated zone, and the saturated zone

2.3.1. Exponential

$$\theta_{exp}(t) = Ae^{-\frac{t}{a}}, t \geq 0 \qquad (4)$$

Where A is the response amplitude, representing the initial response magnitude, a is a shape parameter, and t is time. The exponential IR may be used for drivers or stresses that have a quick and strongly decaying effect on the head, such as precipitation that rapidly recharges a shallow aquifer (Collenteur et al., 2019).

2.3.2. Double Exponential

$$\theta_{dblexp}(t) = A(\frac{1-\alpha}{a_1}e^{-\frac{t}{a_1}} + \frac{\alpha}{a_2}e^{-\frac{t}{a_2}}), t \geq 0 \qquad (5)$$

Where A is response amplitude, alpha determines weight of the two exponential terms. When $\alpha = 0$, the impulse response is dominated by the first exponential decay, and when $\alpha = 1$, it is dominated by the second exponential decay. a_1 and a_2 are constants that control the rate of decay of the impulse response. Adjustments to α can be used to model different response behaviors in hydrological systems, such as contribution to simulated head by fast and slow flow paths through multiple aquifers (Collenteur et al., 2019).

2.3.3. Gamma

$$\theta_{gam}(t) = A\left(\frac{t^{n-1}}{a^n \Gamma(n)}\right)e^{-\frac{t}{a}}, t \geq 0 \tag{6}$$

Where A is the scaling factor, a and n are shape parameters, and Γ is gamma function. This IR, commonly referred to as the scaled gamma distribution (e.g., Besbes and De Marsily, 1984), is often used to simulate a response to areal recharge through the soil column (Collenteur et al., 2019).

2.3.4. One

The "One" model provides an instant response with no lag and a single parameter d, representing the base elevation of the model (Collenteur et al., 2019).

2.3.5. Four Parameter

$$\theta_f = At^{n-1}e^{-\frac{t}{a}-\frac{ab}{t}}, t \geq 0, \tag{7}$$

where A is the scaling parameter, and a, b, and n are shape parameters. When $n > 1$, the function simulates a delayed response in groundwater levels to recharge. For $n \leq 1$ and $b = 0$, groundwater levels respond instantaneously to recharge. If $n = 1$ and $b = 0$, then Eq. (8) reduces to an exponential response Eq. (4) (Collenteur et al., 2019).

2.4. Estimating Recharge

Recharge was one of the primary potential stresses we wanted to investigate in this study. Here, we used two different recharge models: linear, described in Section 2.4.1, and nonlinear, described in Section 2.4.2.

2.4.1. Linear Recharge Model

One common approach to estimate recharge flux in Eq. (3) is through a simple linear function of P precipitation and E_p potential evaporation (e.g., Berendrecht et al., 2003; von Asmuth et al., 2008), but ET_0 can be used in place of E_p (Collenteur et al., 2021) making the equation:

$$R = P - fET_0, \tag{8}$$

where f is a parameter that is calibrated. Because there is no clear interpretation of parameter f, it has been referred to as a crop factor (Berendrecht et al., 2003) or the evaporation factor (Obergfell et al., 2019). It has been noted that the value of f may depend on soil and land cover, and instead of a single crop, it incorporates the "average reduction of the evaporation due to actual soil water shortages" (von Asmuth et al., 2008). Eq. (8) does not account for recharge that may result from storage in the unsaturated zone because it does not have a storage parameter; therefore, it must be accounted for in the IR function (Collenteur et al., 2021). In the linear model, the four-parameter IR characterizes the entire system: the root zone, the unsaturated zone, and the saturated zone. The linear model has six parameters to be estimated: A, n, a, and b of the response function (Eq. 7), the evaporation factor f (Eq. 8), and the base level of the model d (Eq. 1).

2.4.2. Nonlinear Recharge Model

The linear recharge model depends upon the four-parameter IR to simulate the effects of contributions from storage to groundwater recharge. In contrast, the nonlinear model, loosely based on the FLEX framework used for rainfall-runoff modeling (Fenicia et al., 2005), uses a soil-water storage concept to account for the temporal variation of storage of water in the root zone with precipitation and potential evaporation as inputs for the model (detailed in Collenteur et al., 2021). Recharge to the groundwater R is computed using Campbell's approximation for unsaturated hydraulic conductivity (Campbell, 1974):

$$R = k_s \left(\frac{S_r}{S_{r,max}} \right)^{\gamma}, \tag{9}$$

where k_s is saturated hydraulic conductivity, S_r is the amount of water in the root zone reservoir, $S_{r,max}$ is the maximum capacity of the root zone reservoir, and γ is a parameter that determines how nonlinear this flux is with respect to the saturation of the unsaturated zone (Collenteur et al., 2021). Figure 5 illustrates a conceptual model of the nonlinear recharge model.

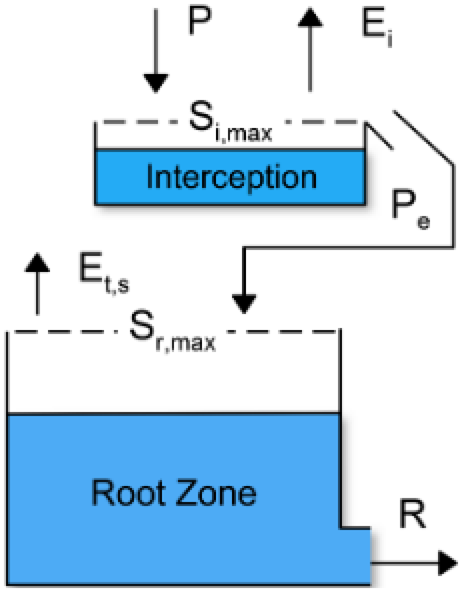


Figure 5. Conceptual model of nonlinear recharge model depicted as two connecting reservoirs: one for interception and the second representing the root zone. P is precipitation, E_i is interception evaporation, $S_{i,max}$ is the maximum capacity of the interception reservoir, $S_{r,max}$ is the maximum capacity of the root zone reservoir, and $E_{t,s}$ is the combined flux of transpiration by vegetation and evaporation in the soil. Adopted from Collenteur et al. (2021).

2.5 Modeling approach

Data preparation, model creation, and parameter optimization for time series analysis were conducted using Pastas, an open-source Python package (Collenteur et al., 2019). Pastas extensively uses Pandas, a Python package adept at managing time series data (McKinney, 2010), requiring all time series to be formatted as Pandas Series or DataFrame objects. Although observed heads can be recorded at varying times, Pastas handles regularly spaced stress series in simulations and can convert irregular time series into regular intervals. Once Pastas is installed, the basic workflow of a Pastas model is as follows:

1. Import groundwater head and stress time series from files and store them as Pandas Series or DataFrame objects.
2. Initiate a model object with the observed head series.
3. Construct StressModel objects with observed stress and a response function, and integrate them into the model.

   a. Another StressModel class, the RechargeModel class, can be used for groundwater recharge effects with rainfall and potential evaporation as inputs. The linear model (rch.Linear) applies Eq. (8) to estimate net groundwater recharge.
   b. Additionally, a nonlinear recharge model (rch.FlexModel) applies Eq. (10) to estimate recharge flux, incorporating precipitation and potential evaporation data.
4. Determine the model's parameters and evaluate fit statistics.
5. Generate visual outputs.
6. Examine residuals and noise.

We estimated all model parameters by fitting simulated groundwater levels to observed data. For our research, we used the Pastas default, which employs a nonlinear least-squares method to minimize the sum of weighted squared noise (e.g., von Asmuth and Bierkens, 2005) for simultaneous parameter estimation in each model (Collenteur et al., 2019; Collenteur et al., 2021). Since the ideal model for each specific stress and monitoring well is not known, we employed a multi-model approach, evaluating the effectiveness of different model configurations, each progressively more complex, to simulate the groundwater levels and understand the stresses that contribute most to groundwater level variation. We used the Kling-Gupta efficiency (Gupta et al., 2009; Kling et al., 2012), or KGE, as our primary method for evaluating model performance. The KGE is related to the widely used Nash-Sutcliffe efficiency (NSE) and provides a metric that integrates correlation, bias, and variability between simulated and observed data and is widely used for hydrologic studies (Zipper et al., 2018; Zipper et al., 2019; Zipper, Gleeson et al., 2021). A KGE of 1.0 indicates perfect agreement between simulations and observations and $KGE < 1$ indicates decreasing model performance, with any $KGE > -0.41$ indicating the model outperforms the mean of the observational data (Knoben et al., 2019). In addition to KGE, we also calculated the root mean squared error (RMSE) for each simulation.

The Appendix provides example code demonstrating the creation of a TFN model using Pastas and visualizing model output.

3. Results

Here, we briefly discuss the overall results of our modeling efforts, and in Sections 3.1 and 3.2, we detail modeling efforts for each monitoring well, including the two best models (e.g., highest KGEs) for each well. As previously mentioned, we began by assessing the performance of simple models that use a single stress and IR to explore drivers of changes in groundwater levels. Tables 2 and 3 summarize our modeling results for 4-6 EIS 2 and 3-5 MOR, respectively. Generally, models for 4-6 EIS 2 produced better fit statistics — including R^2, root-mean-square error (RMSE), and KGE — than those calculated for 3-5 MOR. We observed that models employing a nonlinear approach to estimate recharge generally performed better than those using a linear recharge approach. In particular, simulations with nonlinear recharge models yielded the highest KGE values for each well. However, across all models, there was a consistent underestimation of the largest recharge events, a trend more pronounced in the 3-5 MOR simulations. Furthermore, models simulating groundwater levels in 4-6 EIS 2 showed considerable improvement when stream stage data were included as an input, unlike models that excluded this stress.

Table 2. Results from 4-6 EIS 2 models. IR is impulse response, precip is precipitation, CWB is climatic water balance, and ET_0 is reference ET (evapotranspiration).

Model	**Stresses**	**Stress Model IR**	**Recharge Model**	**Recharge Model IR**	**R^2**	**RMSE**	**KGE**	**Model Summary**
E1	precip	Gam	—	—	0.23	0.22	0.27	Poor fit due to unaccounted contributions
E2	stage	One	—	—	0.61	0.15	0.63	Good fit, indicative of strong relation between N04D stream and well
E3	CWB	FourP	—	—	0.45	0.18	0.54	Moderate fit, likely unaccounted contribution from stream
E4	stage, CWB	One, FourP	—	—	0.65	0.14	0.75	Good fit but does not simulate peaks as well as model E7. **Details in Section 3.1.1.**
E5	precip, ET_0	—	Linear	DblExp	0.43	0.19	0.51	Moderate fit, captures more highs and lows, but overestimates lows
E6	precip, ET_0	—	Non-Linear	DblExp	0.41	0.19	0.36	Poor fit in general
E7	stage, precip, ET_0	One	Non-Linear	Exp	0.67	0.14	0.82	Best fit, best captures peaks, but flat in areas where stage is zero. **Details in Section 3.1.2.**

Table 3. Results from 3-5 MOR models. IR is impulse response, precip is precipitation, CWB is climatic water balance, ET_0 is reference ET (evapotranspiration), and EIS Sim is the simulated head time series.

Model	Stresses	Stress Model IR	Recharge Model	Recharge Model IR	R^2	RMSE	KGE	Model Summary
M1	precip	Gam	—	—	0.06	0.23	-0.10	Very poor fit, resembles precip time series, unaccounted stress
M2	stage	Gam	—	—	0.05	0.23	-0.13	Very poor fit, resembles stage time series, unaccounted stress
M3	CWB	Exp	—	—	0.18	0.22	0.20	Poor fit, underestimates highs and overestimates lows
M4	CWB, stage	Exp, DblExp	—	—	0.21	0.21	0.23	Poor fit, underestimates highs and overestimates lows NOTE: Hantush IR worked the best with these stresses
M5	precip, ET_0	—	Linear	DblExp	0.30	0.20	0.40	Poor fit, underestimates highs, but captures some lows
M6	precip, ET_0	—	Non-Linear	DblExp	0.07	0.23	-0.09	Very poor fit, captures almost nothing, unaccounted stress
M7	precip, ET_0, stage, EIS Sim	Gam, Gam	Linear	DblExp	0.58	0.15	0.65	Captures seasonal and downward trend in diffuse recharge and indicates that the primary contributor is "leaky aquifer" recharge. **Details in Section 3.2.1.**
M8	precip, ET_0, EIS Sim	DblExp, DblExp	Non-Linear	Gamma	0.58	0.15	0.66	Likely the best model as it makes the most physical sense. **Details in Section 3.2.2.**

3.1. 4-6 EIS 2 models

The KGE for models that simulated groundwater levels in 4-6 EIS 2 (table 2) ranged between 0.27 and 0.82. Unsurprisingly, our simplest model (table 2, model E1), where only a gamma IR was used to represent the effects of precipitation with no other stresses, yielded the lowest KGE for predicting changes in groundwater levels. Conversely, when the gamma IR was used to characterize groundwater response to stage (table 2, model E2), there was a marked improvement in model performance: R^2 increased from 0.23 to 0.61 and KGE increased from 0.27 to 0.63. This model configuration produced the best-fit statistics across all our single-stress models (including 3-5 MOR). The performance of model E2 was not surpassed until we developed model E4, the model with the second-best-fit statistics (table 2). The following subsection headers name the well from which the observations are being simulated, the stresses being used as inputs, recharge model (if applicable), and the IR that is being used to characterize the response and/or recharge. For example: X.X.X. Model number — Well name: stress(IR) + stress(IR) + Recharge model(precip, ET_0 (IR))

3.1.1. Model E4 — 4-6 EIS 2: stage(One) + CWB(FourP)

Model E4 (see table 2), our second-best-fitting model, exhibited reasonably good performance in simulating groundwater levels at 4-6 EIS 2 and had a strong agreement with observed data (KGE = 0.76), though it tended to underestimate the highest head observations, suggesting it may not be capturing the full magnitude of large recharge events (fig. 6a). This model represented the combined effects of stage with an IR that simulates an immediate response with no lag (fig. 6b) and the effect of CWB through an IR that is used to characterize a lagged response (fig. 6c). The maximum contributions of stage to head variability (up to ~1.0 m) were approximately double the maximum contributions of recharge (up to ~0.5 m). This model accurately simulated the available observed groundwater levels from the latter half of 2015 until early 2017 but underestimated a major recharge event towards the end of 2015. During the period from 2017 to mid-2018, the model generally underestimated observed groundwater levels, including a larger event in the first quarter of 2017. In the third quarter of 2018, the model overestimates the observed levels. The model's accuracy improved at the beginning of 2019, closely matching the observed levels, but underestimates the larger recharge event in the second quarter of 2019. From mid-2019 to the end of the observation period, the model tended to overestimate the observed groundwater levels.

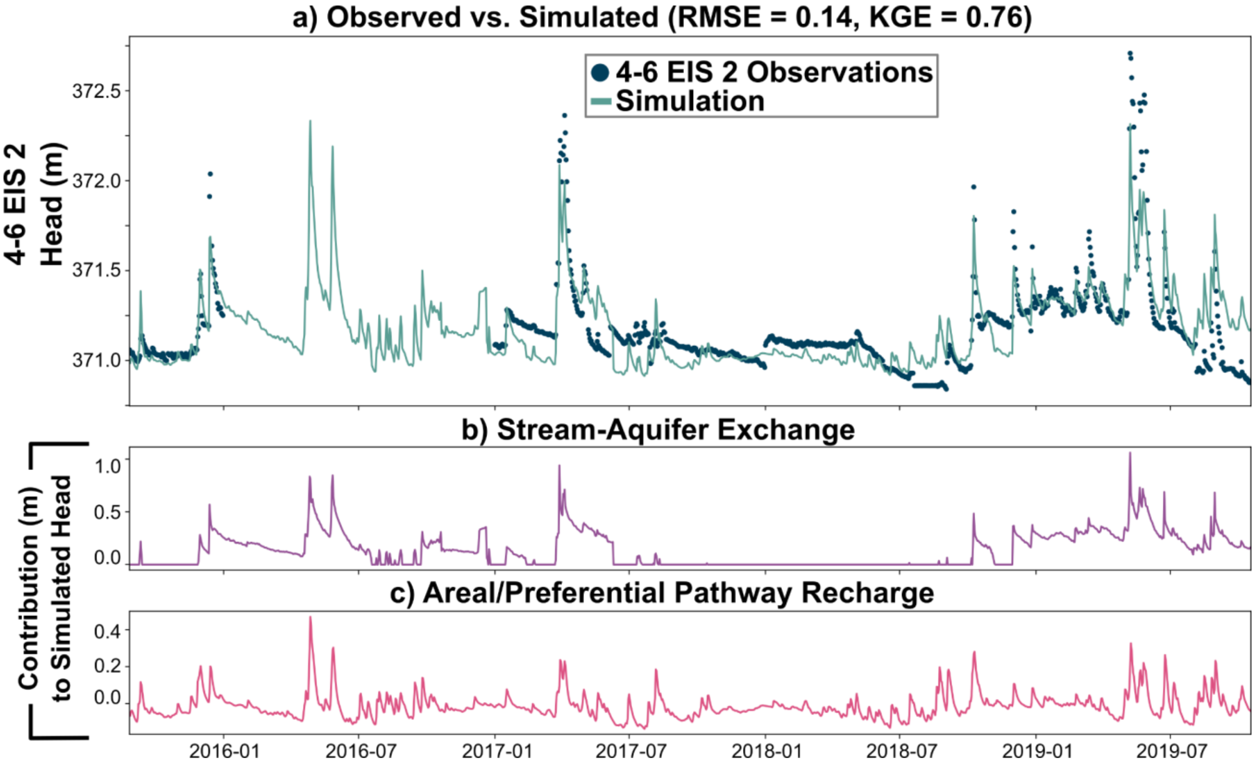


Figure 6. Results from the second-best-fitting model (E4) for 4-6 EIS 2. a) Observed (blue) and simulated (green) head time series. b) Contribution from stream stage to simulated head, representative of exchange between N04D stream and the upper Eiss Member (from 4-6 EIS 2 head). c) Contribution from areal recharge and preferential pathway recharge to simulated head.

3.1.2. Model E7 — 4-6 EIS 2: stage(One) + FlexModel(precip, ET_0 (Exp))

Model E7 (see table 2) delivered the most accurate simulation (quantified by the aforementioned goodness-of-fit metrics) for the 4-6 EIS 2 groundwater levels (fig. 7a), where it reliably captured major recharge events but presented a less consistent fit throughout the entire observation period. This model, which employed the same stage characterization as model E4 (fig. 7b), differed in that it used a nonlinear recharge model with an exponential IR (used to characterize rapid responses, often in shallow aquifers) to better estimate areal/preferential pathway recharge that is likely more realistically mechanistic. As a result, model E7 tended to have more rapid peaks and recessions in the areal/preferential pathway recharge than model E4 (fig. 7c). The contributions to head variability from stage (~1.0 m) and recharge (~0.5 m) were approximately the same magnitude as model E4. From the onset of the observations until early 2017, the model significantly underestimated groundwater levels, with the exception of the first major event, which it modeled with relative accuracy. Despite this initial underperformance, the model accurately represented the major recharge event in the first quarter of 2017. Due to the more rapid response to recharge, this model was unable to simulate variability in groundwater levels during periods when the stream was dry and there was no recharge from mid-2017 to mid-2018 (fig. 7a). The model accurately rendered the peak of another significant recharge event in the fourth quarter of 2018, but it failed to capture smaller recharge events occurring from late 2018 through the second quarter of 2019. Nevertheless, the model effectively simulated the largest recharge event in the second quarter of 2019. After this event, the model consistently overestimated groundwater levels for the remainder of the observation period. This pattern suggests that although model E7 can effectively simulate high-magnitude events, it appears to encounter limitations at approximately 371 meters, suggesting a potential threshold effect influencing its performance. Therefore, although the model has the best-fit statistics of all the wells for 4-6 EIS 2 (table 2), it may not provide physically realistic predictions across the full range of observed groundwater conditions.

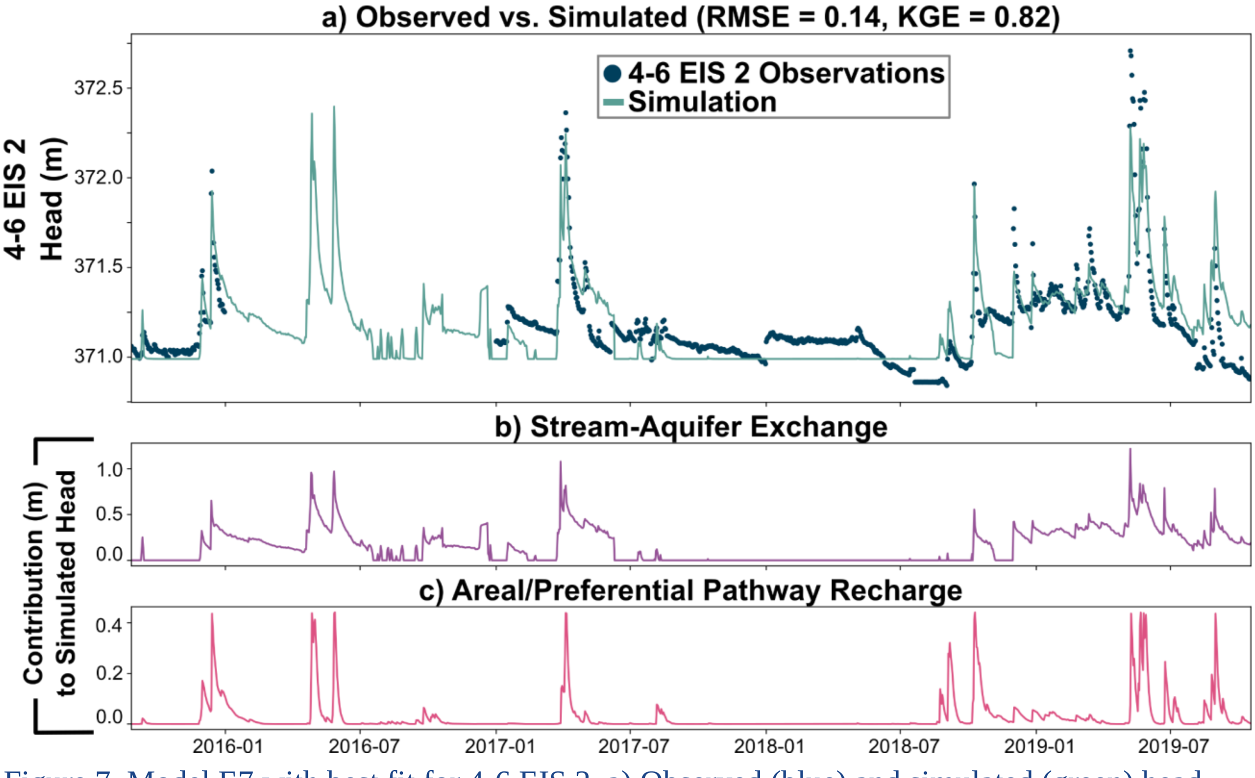


Figure 7. Model E7 with best fit for 4-6 EIS 2. a) Observed (blue) and simulated (green) head time series. b) Contribution from stream stage to simulated head, representative of exchange between N04D stream and the upper Eiss (from 4-6 EIS 2 head). c) Contribution from areal recharge to simulated head.

3.1.3. Hindcasting head in 4-6 EIS 2

The 3-5 MOR monitoring well is screened in the Morrill Limestone Member, which underlies the Eiss Limestone Member. Previous studies suggest that the Eiss may function as a leaky aquifer, with drainage flowing into the Morrill and thus influencing the groundwater levels within 3-5 MOR (e.g., Sullivan et al., 2019). Recognizing the significance of this interaction for the groundwater dynamics in 3-5 MOR, it was imperative to quantify the contribution of the upper Eiss to recharge in the Morrill, but 4-6 EIS 2 data were not available prior to 2015 (fig. 3). To evaluate the potential importance of this contribution, we adapted the model configuration from model E7 in table 2. Integrating historical data on stage, precipitation, and ET_0, we generated a time series of head in 4-6 EIS 2, which originally only covered the period from 2015 to 2020, that is hindcasted back to 2004 to match the period of observational data in 3-5 MOR. These hindcasted data were standardized by subtracting the mean to produce the timeseries shown in fig. 8, which was used as a stress input for a TFN model designed to simulate the groundwater levels in 3-5 MOR. The hindcasted time series of 4-6 EIS 2 exhibited increased variability between 2004 and 2011. Notably, it featured a pronounced spike in 2008, which appears to be an anomaly and is likely erroneous as it is substantially outside the range of variability observed elsewhere and, based on the observed well depths (fig. 2), would indicate flooding at the land surface. The exceptionally large spike, while conspicuous, is likely due to an unusually high stage observation (fig. 4d).

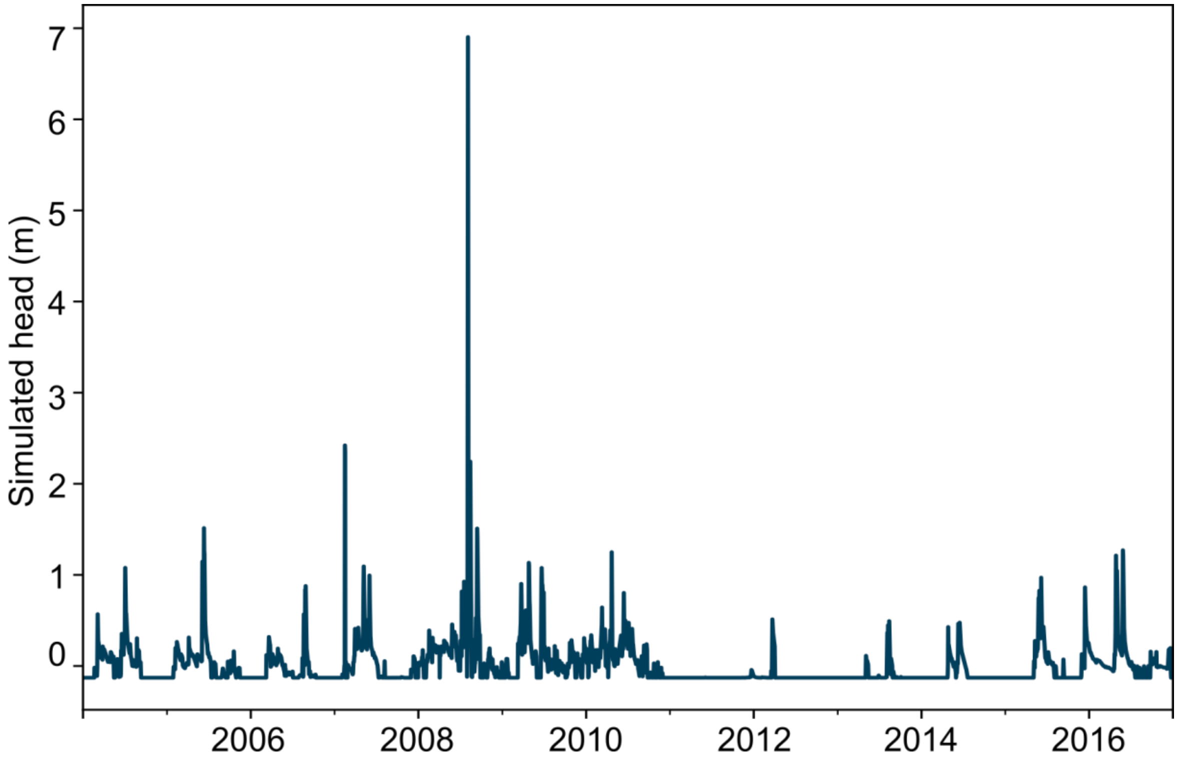


Figure 8. 4-6 EIS 2 simulated head time series was generated using parameters from model E7. Data were standardized by subtracting the mean of the time series. Note: The exceptionally large spike, while conspicuous, was not identified as an outlier in our data preparation process.

3.2. 3-5 MOR models

Groundwater levels in 3-5 MOR had a more cyclic nature with seasonal rises and falls in the water table, unlike the flashy water level variation in 4-6 EIS 2, suggesting different processes may be controlling groundwater dynamics. The KGE for models that simulated groundwater levels in 3-5 MOR, as outlined in table 3, exhibited a wide range from -0.13 to 0.66. Among these, model M2 (referenced in table 3) resulted in the lowest KGE. This model exclusively used a gamma IR to depict the influence of stream stage on the groundwater levels in 3-5 MOR. Notable model performance improvement was first achieved with model M5 (table 3), although its KGE still remained comparatively low, especially when contrasted with the more rapid improvements seen in the models for 4-6 EIS 2 (table 2, model E2; KGE = 0.63). In model M5 (table 3; KGE = 0.40), a linear recharge model was employed to estimate and simulate recharge, yielding better goodness-of-fit statistics relative to the initial four models. Further improvements in model performance for 3-5 MOR were observed when the EIS simulation was incorporated as a stress, first used in the case of the second-best-fitting model, M7 (table 3).

3.2.1. Model M7 — 3-5 MOR: rch.Linear(precip, ET_0 (DblExp)) + stage(Gam) + EIS simulated head(Gam)

Model M7, our second-best-fitting model for 3-5 MOR (table 3), demonstrated a reasonable ability to simulate groundwater levels (fig. 9a), though it did not successfully simulate the magnitude of larger recharge events. The model's configuration employed a linear recharge model with a double exponential IR that can be used to characterize exchange in multi-aquifer systems to simulate areal recharge that effectively captures both a seasonal fluctuation and a long-term declining trend in areal recharge (fig. 9b). Also, the linear recharge model was able to replicate signals that appeared to be generated from infiltration through preferential pathways, as indicated by the small fluctuations across the graph. Additionally, we estimated the contribution from exchange between the stream and 3-5 MOR using a gamma IR, which is often used to simulate lagged responses in systems. Finally, to simulate the potential flow from the upper Eiss Limestone Member to the Morrill Limestone Member, we incorporated the simulation from 4-6 EIS 2 as a stress in the model, also using a gamma IR. The modeled influence of 4-6 EIS 2 over water levels in 3-5 MOR was smoothed and lagged (fig. 9d) relative to the 4-6 EIS 2 groundwater levels used as input (fig. 8).

Analyzing the model's performance over time, it accurately simulated the seasonal rise and fall of the water table but did not capture the significant recharge events that typically occurred in the spring of each year. Although minimum groundwater levels were accurately predicted from 2007 to 2009, the model overestimated minimum groundwater levels in 2010 and underestimated minimum groundwater levels in 2011 and continued to miss the major recharge events, including the largest recharge event in 2013. Between mid-2013 and early 2014, the model overestimated the rising limb of groundwater levels and failed to simulate the subsequent major recharge event in the spring of 2014. From 2015 through the end of 2016, the model attempted to capture the complex dynamics of the 3-5 MOR observations but overestimated the receding limb of observed groundwater levels between late 2015 and early 2016 and underestimated groundwater levels in mid-2016.

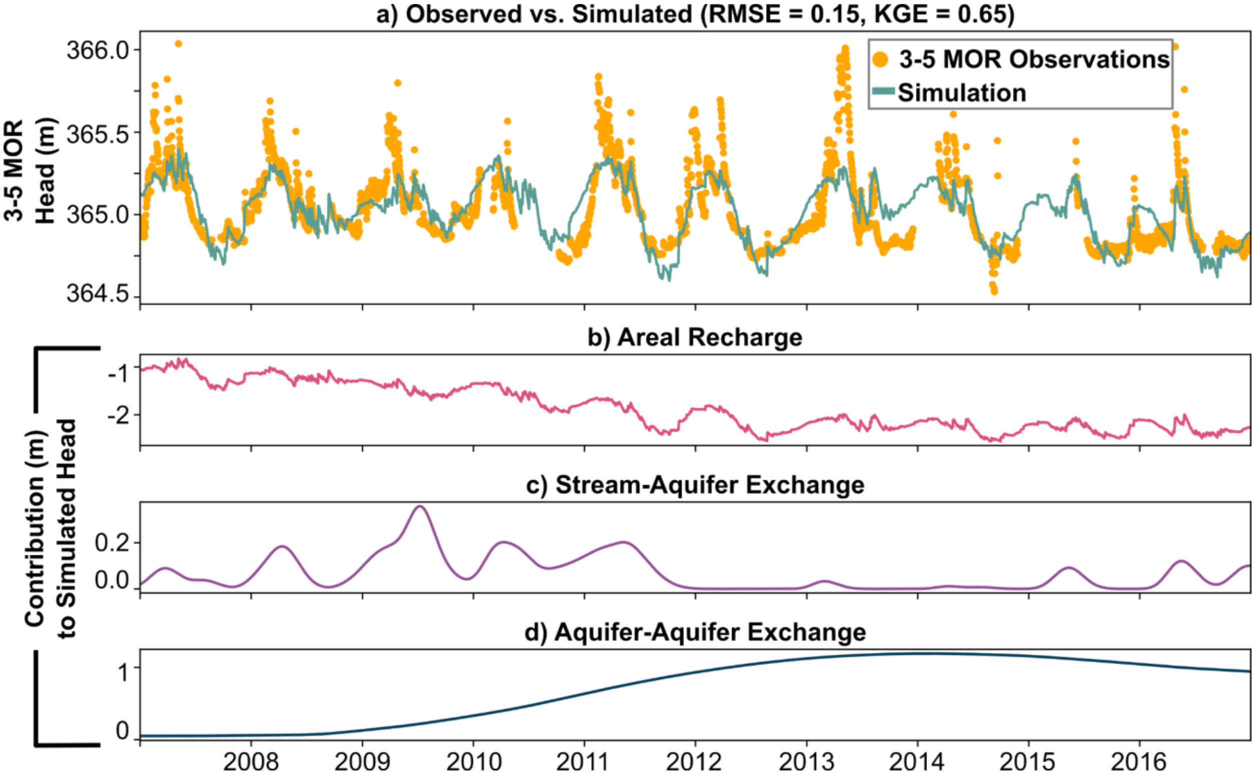


Figure 9. Results from the second-best-fitting model (M7) for 3-5 MOR. a) Observed (yellow) and simulated (green) head time series. b) Contribution from areal recharge using linear model. c) Reference ET component of contribution from areal recharge using linear model. d) Contribution from exchange between N04D stream and the Morrill. d) Contribution from exchange between the Morrill and the upper Eiss (from 4-6 EIS 2 head).

3.2.2. Model M8 — 3-5 MOR: FlexModel(precip, evap(Gamma)) + precip(DoubleExponential) + EIS Simulated Head(DoubleExponential)

Model M8 provided a simulation (fig. 10a) that closely resembled that of model M7 but displayed subtle yet critical differences (table 3). First, model M8 adopted a nonlinear recharge model with a gamma IR that depicted a pronounced seasonal pattern in areal recharge, with the most notable oscillations occurring from mid-2009 to early 2013 (fig. 10b). Second, model M8 simulated preferential pathway recharge by integrating precipitation data directly into the model using a double exponential IR (fig. 10c) in addition to integrating the role of precipitation in the climatic water balance term. Finally, the impact of the potential flow from the Eiss Limestone Member on groundwater levels in the Morrill Limestone Member was estimated using a double exponential IR (fig. 10d). Overall, model M8 demonstrated a higher capacity to simulate lower groundwater level observations when compared to model M7. Nevertheless, it encountered similar difficulties in accurately re-creating large recharge events. From 2007 to 2011, model M8 provided a more precise representation of lower groundwater levels than model M7 but still underestimated observed levels from 2011 to mid-2012. Between 2013 and late 2015, the model closely matched the groundwater level observations, aside from an overestimation in early 2016. After this, the model persistently overestimated the observed data until the end of the observation period. Interestingly, although both M7 and M8 showed a relatively muted response to water level variations from 4-6 EIS 2, the maximum contributions in model M7 occurred between 2013 and 2016, when model M8 showed minimum contributions of 4-6 EIS 2 to 3-5 MOR.

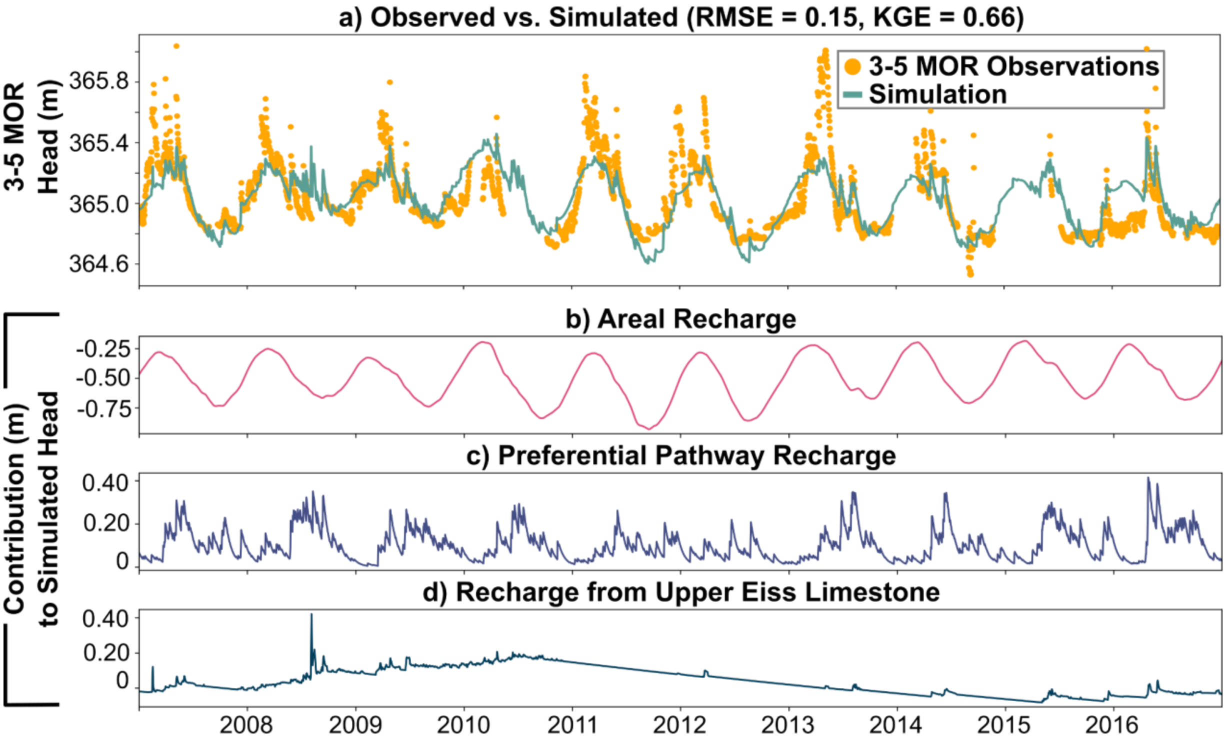


Figure 10. Model M8 with best fit for 3-5 MOR. a) Observed (yellow) and simulated (green) head time series. b) Contribution from areal recharge using nonlinear model. c) Contribution from recharge through preferential flow paths. d) Contribution from exchange from the upper Eiss Limestone Member (from 4-6 EIS 2 head).

4. Discussion

Our study compared different TFN models to simulate groundwater level variation and estimate recharge and individual contributions to changes in groundwater levels for wells in two different limestone units. The models for 4-6 EIS 2 and 3-5 MOR showcased their ability in estimating groundwater level variation in a complex hydrogeological setting and groundwater response to multiple processes, including links between surface water and groundwater, at Konza

Prairie. However, the models' inability to accurately represent some prominent recharge events, including typical annual spring recharge at 3-5 MOR, indicates that in this setting they may still have limitations in accurately representing large and rapid recharge events. Below, we discuss the implications of these findings for our understanding of groundwater dynamics in the studied area, how these findings align with existing hydrological theories on groundwater-surface water interactions in Konza, potential improvements that could be made to the models, and the ability of relatively simple models, such as TFN models, to provide insight in complex settings such as Konza Prairie.

4.1. 4-6 EIS 2 as main source of streamflow

Our findings are consistent with prior studies that showed significant exchange between the Eiss and streamflow in the watershed N04D at Konza Prairie. Past work revealed that stream water at this site is a mixture of four different sources: groundwater flow through carbonate aquifers, groundwater flow along carbonate-mudstone contacts, surface runoff, and soil water (Sullivan et al., 2019). Additionally, during stable isotope analysis conducted between 2010 and 2017, Keen et al. (2023) discovered "substantial isotopic overlap between stream and groundwater sources" strongly suggesting that groundwater input is an important source for streamflow in the area. Moreover, endmember mixing analysis results demonstrate that a vast majority (96.3%) of stream discharge is attributed to groundwater, with the upper Eiss alone accounting for 73.5% of this contribution (Hatley et al., 2023). Although our models might suggest that the groundwater levels in 4-6 EIS 2 are driven by stream-aquifer exchange, this past work indicates that it is more likely the Eiss and stream are both driven by precipitation and the Eiss has a critical role in sustaining streamflow in the Konza Prairie. Since our best-fitting models for the 4-6 EIS 2 used a "One" IR for stream stage, which indicates a direct and immediate link between stream stage and groundwater levels, they do not provide information about the direction of flow between the stream and aquifer. Nonetheless, our models underscore the significant interconnection between the precipitation, stream stage and groundwater levels in the Eiss. This interpretation is supported by the consistent patterns observed in hydrological data and corroborative findings from isotopic and mixing analyses (e.g., Hatley et al., 2023; Sullivan et al., 2019). Although groundwater is often thought to be a slow contributor to streamflow, the rapid interactions we observe between the stream stage and groundwater levels (fig. 7) are consistent with recent findings of relatively short groundwater residence times and a high fraction of water less than three months old in the streams despite the high groundwater fraction (Swenson et al., 2024).

The relationship between precipitation, streamflow, and groundwater levels in 4-6 EIS 2 is also evident from the alignment of peaks in the precipitation and stage time series with those in the groundwater levels, potentially indicating that precipitation is a common driver of both streamflow and groundwater fluctuations (fig. 11). In model E6 (table 2), the areal/preferential recharge demonstrates numerous but smaller responses to precipitation events compared to model E7 (fig. 11b). This can be explained by the functional form and parameterization of the IR used in these models. The parameterization of model E6, where $n < 0$ and $b \approx 0$, leads to an immediate response from the CWB without the lag that is often characterized with a four-parameter IR (see Section 2.3.5). The immediate response of the groundwater system is indicative of the model's sensitivity to changes in meteorological conditions. Model E7, however, presents a different recharge pattern that exhibits fewer but more substantial responses to precipitation (fig. 11a in green), which suggests model E7 buffers meteorological conditions through the soil column to create a lag between meteorological conditions and groundwater recharge. Though model E7 employs an exponential IR that also captures a rapid response, the difference between models arises through the implementation of a nonlinear recharge model in E7. Model E7 performs slightly better at simulating peaks in the groundwater hydrograph, but model E6 better simulates groundwater dynamics during recession. This suggests that a model form that combines these two characteristics may be able to simulate the full range of observed groundwater level variation at the site, with relative simplicity when compared to the process of developing numerical models to simulate these processes. However, when Long and Mahler (2013) used lumped-parameter models to estimate groundwater recharge in a karst environment, their models required several types of IRs to simulate their observations, highlighting the complexity of modeling hydrological processes in karst systems.

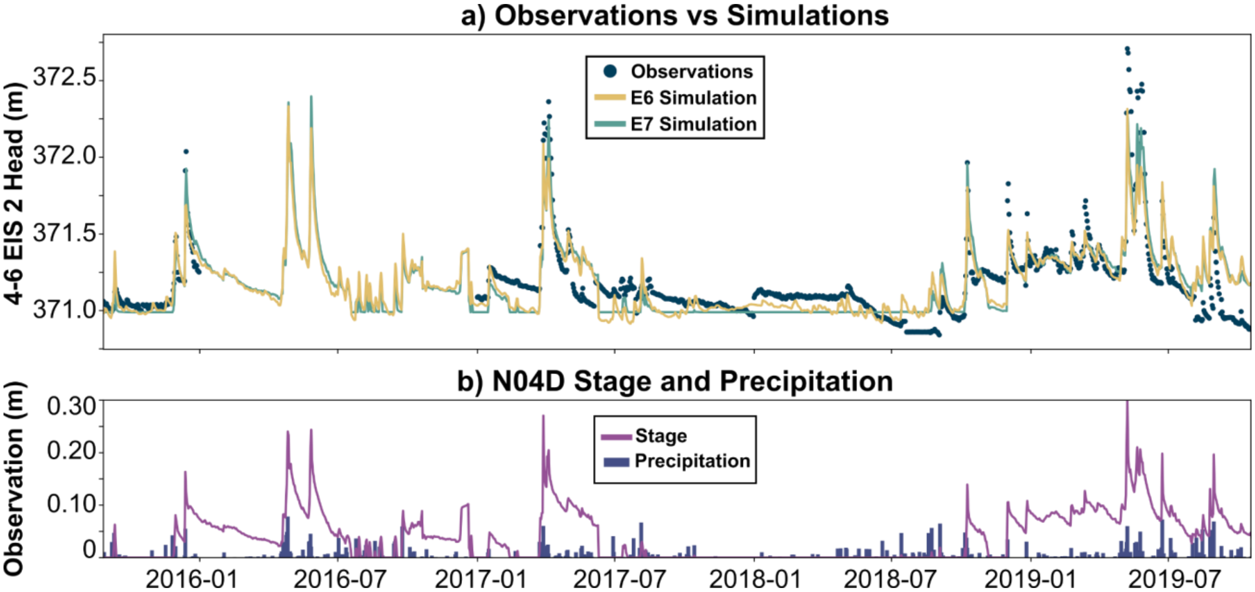


Figure 11. Comparison of responses from simulations and estimated recharge from two best 4-6 EIS 2 models from stage and precipitation stresses. a) Comparison of simulations from models E6 and E7 for 4-6 EIS 2. b) Stage and precipitation stress time series.

4.2. Complex relations between 4-6 EIS 2 and 3-5 MOR

Recent research by Hatley et al. (2023) provides evidence that, in the limestone aquifers of the Konza Prairie, vertical connections between different geological units can result in overlying hydrostratigraphic units “leaking” into underlying units. This phenomenon is supported by the relationship we observe between the upper Eiss Limestone Member and the Morrill Limestone Member. The Morrill likely receives continuous groundwater input from the overlying upper Eiss, especially after storm events when water levels in the Eiss and the stream are high. This water may either flow through the surface stream network from the upper Eiss before infiltrating into the Morrill, or there may be subsurface connections between the two, for example through fractured mudstone. Water levels in the Morrill then recede relatively slowly compared to in the Eiss (fig. 10). Although the 3-5 MOR well receives substantial recharge directly from the stream, as indicated by Barry (2018) and Macpherson (1996), the complexity of these interactions poses challenges in correlating stream stage variations with groundwater levels in the 3-5 MOR. The measured stage at the N04D outlet flume, which is hundreds of meters downstream from these wells, typically shows zero-flow from July to February (far left diagram in fig. 12), contrasting with locations of near-perennial water presence at unit contacts upstream near these monitoring wells (Macpherson, 1996). This flow pattern begins where the upper Eiss transitions to the lower Eiss and ends where the Stearns Shale transitions to the Morrill Limestone Member.

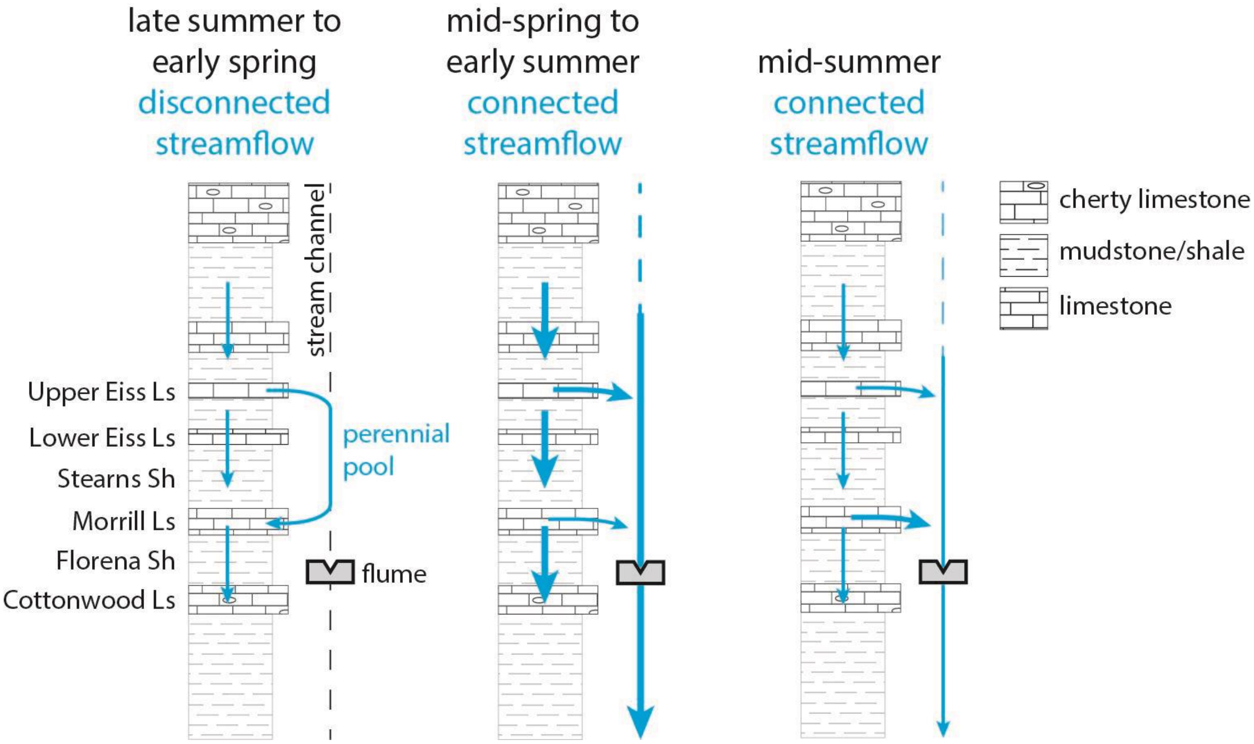


Figure 12. Representation of important hydrostratigraphic units and surface/subsurface flowpaths during connected and disconnected streamflow in watershed N04D at Konza Prairie.

This suggests a seasonal flow path in which water from the upper Eiss travels a certain distance in the stream channel before re-infiltrating and recharging the Morrill. Streamflow likely infiltrates the Morrill before reaching the flume (e.g., Hatley et al., 2023; Sullivan et al., 2019), leading to minimal correlation between stream stage and groundwater levels in 3-5 MOR since the stream stage data we have are not representative of water levels in the stream immediately proximal to these monitoring wells. This lack of correlation complicates efforts to accurately assess the effect of stream stage changes on groundwater levels and determine causal relationships between precipitation, stream stage, and groundwater levels. Incorporating the 4-6 EIS 2 simulation into our models for 3-5 MOR not only enhances model performance but also provides a more physically representative model. Using 4-6 EIS 2 as input into the 3-5 MOR model may also provide a proxy for the impacts of the stream on groundwater levels in 3-5 MOR.

4.3. Modeling considerations

One of the primary challenges in hydrological modeling is the issue of equifinality. Equifinality refers to the phenomenon in which different combinations of model parameters can result in equally plausible simulations of observed hydrological data. This situation often arises when the effects of one process, such as groundwater pumping, are counterbalanced by another, such as evaporation (Shapoori et al., 2015). Such compensatory dynamics can yield good model fits but may not accurately represent the underlying physical processes. This issue becomes particularly critical when model parameters are intended to quantify specific physical quantities, such as groundwater recharge (Beven, 1993). The risk is that models might provide a false sense of accuracy, overlooking the true drivers of hydrological changes. By gradually adding complexity and testing alternative model configurations through the use of multiple stresses and Irs, our approach identified and discussed plausible mechanisms for potential equifinality.

Improving model performance and estimating error would be useful for future work. Reducing autocorrelation would be the first challenge to overcome. In the TFN framework, estimates of standard error can be made when autocorrelation is minimized using appropriate noise models (Collenteur et al., 2021), potentially providing an approach to quantify uncertainty. Any future improvements in our models should include the minimization of autocorrelation within the observation series. This can often be accomplished by changing the time steps of an observation time series by removing observations (Collenteur et al., 2021). Additionally, model improvement may be accomplished by considering other nonlinear processes that characterize intermittent surface waters by allowing head response to be a function of itself (e.g., Knotters and De Gooijer, 1999).

Additionally, the model complexity and system linearity are other crucial considerations in hydrological modeling. Most models, including TFN models, perform better when relationships between inputs and outputs are relatively linear (Obergfell et al., 2019). If a hydrological system exhibits significant nonlinear characteristics, such as hysteresis and stability (Zipper, Popescu et al., 2022), standard modeling approaches based on linear superposition of stresses might fail to capture these complex interactions, leading to inaccurate predictions and assessments. In this study, we found that using an exponential IR yielded results similar to those obtained with a more complex four-parameter IR, similar to findings by Collenteur et al. (2021).

Also, we found that using fewer stresses yielded results similar to models that used more. In model M7 for 3-5 MOR, we incorporated four stresses, whereas in model M8, we incorporated only three stresses: both produced models with better goodness-of-fit statistics than the other models. This finding suggests that simpler models can sometimes approximate the behavior of more complex systems without significant loss of accuracy. Overall, our work demonstrates the utility of relatively simple TFN models, especially in complex settings such as the merokarst of Konza Prairie where process-based finite difference or finite-element approaches may not have sufficient resolution to capture fracture flow.

5. Conclusions

Here, we explored hydrological processes in Konza Prairie using TFN models to provide insights into both model selection and hydrological processes. Despite the inherent complexities of the underlying hydrological systems, the TFN models provided reasonably accurate simulations of groundwater level variation in both the Eiss and Morrill Members, demonstrating their capability to estimate recharge and quantify the contributions of different stresses to groundwater levels. Our models suggest that precipitation may be a joint driver of both stream levels and groundwater recharge to the upper Eiss, while in the deeper Morrill unit, our models suggest that diffuse recharge, recharge via preferential pathways, and exchange between the two aquifers influence groundwater level variation. The efficacy of TFN models in deciphering the roles of the Eiss and Morrill, and their interplay with surface water dynamics, offers a clearer understanding of how different hydrological drivers interact within the system. By adequately capturing the vertical "leaking" effect between limestone layers and distinguishing between contributions to groundwater recharge from stream-aquifer interactions and preferential pathway recharge, these models spotlight the potential drivers of groundwater level variation and, ultimately, stream intermittency at Konza Prairie. Since modeling hydrological processes in karst environments is notoriously challenging, TFN models may provide a robust framework for researchers and water resource managers. This approach enables the identification of key hydrological processes and the development of more effective water management strategies, which is crucial in the face of growing environmental challenges and climatic variability.

6. References

Bakker, M., and Schaars, F., 2019, Solving groundwater flow problems with time series analysis: You may not even need another model: Groundwater, v. 57, no. 6, p. 826–833.

Barry, E. R., 2018, Characterizing groundwater flow through merokarst, Northeast Kansas, USA: M.S. thesis, University of Kansas.

Berendrecht, W. L., Heemink, A. W., van Geer, F. C., and Gehrels, J. C., 2003, Decoupling of modeling and measuring interval in groundwater time series analysis based on response characteristics: Journal of Hydrology, v. 278, no. 1, p. 1–16.

Besbes, M., and De Marsily, G., 1984, From infiltration to recharge: Use of a parametric transfer function: Journal of Hydrology, v. 74, no. 3, p. 271–293.

Beven, K., 1993, Prophecy, reality and uncertainty in distributed hydrological modelling: Advances in Water Resources, v. 16, no. 1, p. 41–51.

Briggs, J. M., Knapp, A. K., Blair, J. M., Heisler, J. L., Hoch, G. A., Lett, M. S., and McCarron, J. K., 2005, An ecosystem in transition: Causes and consequences of the conversion of mesic grassland to shrubland: BioScience, v. 55, no. 3, p. 243–254.

Busch, M. H., Costigan, K. H., Fritz, K. M., Datry, T., Krabbenhoft, C. A., Hammond, J. C., Zimmer, M., Olden, J. D., Burrows, R. M., Dodds, W. K., Boersma, K. S., Shanafield, M., Kampf, S. K., Mims, M. C., Bogan, M. T., Ward, A. S., Rocha, M. P., Godsey, S., Allen, G. H. … Allen, D. C., 2020, What's in a name? Patterns, trends, and suggestions for defining non-perennial rivers and streams: Water, v. 12, no. 7, p. 1,980.

Campbell, G. S., 1974, A simple method for determining unsaturated conductivity from moisture retention data: Soil Science, v. 117, no. 6, p. 311.

Collenteur, R. A., Bakker, M., Caljé, R., Klop, S. A., and Schaars, F., 2019, Pastas: Open source software for the analysis of groundwater time series: Groundwater, v. 57, no. 6, p. 877–885.

Collenteur, R. A., Bakker, M., Klammler, G., and Birk, S., 2021, Estimation of groundwater recharge from groundwater levels using nonlinear transfer function noise models and comparison to lysimeter data: Hydrology and Earth System Sciences, v. 25, no. 5, p. 2,931–2,949.

Costigan, K. H., Daniels, M. D., and Dodds, W. K., 2015, Fundamental spatial and temporal disconnections in the hydrology of an intermittent prairie headwater network: Journal of Hydrology, v. 522, p. 305–316.

Datry, T., Boulton, A. J., Bonada, N., Fritz, K., Leigh, C., Sauquet, E., Tockner, K., Hugueny, B., and Dahm, C. N., 2018, Flow intermittence and ecosystem services in rivers of the Anthropocene: Journal of Applied Ecology, v. 55, no. 1, p. 353–364.

Datry, T., Larned, S. T., and Tockner, K., 2014, Intermittent rivers: A challenge for freshwater ecology: BioScience, v. 64, no. 3, p. 229–235.

Datry, T., Truchy, A., Olden, J. D., Busch, M. H., Stubbington, R., Dodds, W. K., Zipper, S., Yu, S., Messager, M. L., Tonkin, J. D., Kaiser, K. E., Hammond, J. C., Moody, E. K., Burrows, R. M., Sarremejane, R., DelVecchia, A. G., Fork, M. L., Little, C. J., Walker,

R. H. … Allen, D.,, 2023, Causes, responses, and implications of anthropogenic versus natural flow intermittence in river networks: BioScience, v. 73, no. 1, p. 9–22.

Dodds, W. K. 2023, ASD02 Stream discharge measured at the flumes on watershed N04D at Konza Prairie: Environmental Data Initiative. http://dx.doi.org/10.6073/pasta/43cac10952a6115c1df8882fa2134a7c

Dodds, W. K., Gido, K., Whiles, M. R., Fritz, K. M., and Matthews, W. J., 2004, Life on the edge: The ecology of Great Plains prairie streams: BioScience, v. 54, no. 3, p. 205–216.

Dodds, W. K., and Oakes, R. M., 2006, Controls on nutrients across a prairie stream watershed: Land use and riparian cover effects: Environmental Management, v. 37, no. 5, p. 634–646.

Dodds, W. K., and Oakes, R. M., 2008, Headwater influences on downstream water quality: Environmental Management, v. 41, no. 3, p. 367–377.

Dodds, W. K., Ratajczak, Z., Keen, R. M., Nippert, J. B., Grudzinski, B., Veach, A., Taylor, J. H., and Kuhl, A., 2023, Trajectories and state changes of a grassland stream and riparian zone after a decade of woody vegetation removal: Ecological Applications, v. 33, no. 4, e2830.

Fenicia, F., Savenije, H. H. G., Matgen, P., and Pfister, L., 2005, Is the groundwater reservoir linear? Learning from data in hydrological modelling: Hydrology and Earth System Sciences Discussions, v. 2, no. 4, p. 1,717–1,755.

Gupta, H. V., Kling, H., Yilmaz, K. K., and Martinez, G. F., 2009, Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling: Journal of Hydrology, v. 377, no. 1, p. 80–91.

Hammond, J. C., Zimmer, M., Shanafield, M., Kaiser, K., Godsey, S. E., Mims, M. C., Zipper, S., Burrows, R. M., Kampf, S. K., Dodds, W., Jones, C. N., Krabbenhoft, C. A., Boersma, K. S., Datry, T., Olden, J D., Allen, G. H., Price, A. N., Costigan, K., Hale, R. … Allen, D. C.,, 2021, Spatial patterns and drivers of nonperennial flow regimes in the contiguous United States: Geophysical Research Letters, v. 48, no. 2, e2020GL090794.

Hatley, C. M., Armijo, B., Andrews, K., Anhold, C., Nippert, J. B., and Kirk, M. F., 2023, Intermittent streamflow generation in a merokarst headwater catchment: Environmental Science: Advances, v. 2, no. 1, p. 115–131.

Hayden, B. P., 1998, Grassland Dynamics: Long-Term Ecological Research in Tallgrass Prairie: New York, Oxford University Press. p. 19–34.

Jemeļjanova, M., Collenteur, R. A., Kmoch, A., Bikše, J., Popovs, K., and Kalvāns, A., 2023, Modeling hydraulic heads with impulse response functions in different environmental settings of the Baltic countries: Journal of Hydrology: Regional Studies, v. 47, p. 101416.

Katz, G. L., Denslow, M. W., and Stromberg, J. C., 2012, The Goldilocks effect: Intermittent streams sustain more plant species than those with perennial or ephemeral flow: Freshwater Biology, v. 57, no. 3, p. 467–480.

Keen, R. M., Nippert, J. B., Sullivan, P. L., Ratajczak, Z., Ritchey, B., O'Keefe, K., and Dodds, W. K., 2023, Impacts of riparian and non-riparian woody encroachment on tallgrass prairie ecohydrology: Ecosystems, v. 26, no. 2, p. 290–301.

Kirk, M., and Macpherson, G. L., 2023, AGW03 Konza prairie long-term high frequency groundwater level and temperature from wells on N04d: Environmental Data Initiative. http://dx.doi.org/10.6073/pasta/6faaf55025f355358510260e0b9e4fb7

Kling, H., Fuchs, M., and Paulin, M., 2012, Runoff conditions in the upper Danube basin under an ensemble of climate change scenarios: Journal of Hydrology, v. 424–425, Supplement C, p. 264–277.

Knoben, W. J. M., Freer, J. E., and Woods, R. A., 2019, Technical note: Inherent benchmark or not? Comparing Nash–Sutcliffe and Kling–Gupta efficiency scores: Hydrology and Earth System Sciences, v. 23, no. 10, p. 4,323–4,331.

Knotters, M., and De Gooijer, J. G., 1999, TARSO modeling of water table depths: Water Resources Research, v. 35, no. 3, p. 695–705.

Krabbenhoft, C. A., Allen, G. H., Lin, P., Godsey, S. E., Allen, D. C., Burrows, R. M., DelVecchia, A. G., Fritz, K. M., Shanafield, M., Burgin, A. J., Zimmer, M. A., Datry, T., Dodds, W. K., Jones, C. N., Mims, M. C., Franklin, C., Hammond, J. C., Zipper, S., Ward, A. S. … Olden, J. D., 2022, Assessing placement bias of the global river gauge network: Nature Sustainability, v. 5, no. 7, p. 586–592.

Lewis, M. A., Cheney, C. S. and O Dochartaigh, B. E., Guide to permeability indices: British Geological Survey, 2006, p. 29.

Long, A. J., and Mahler, B. J., 2013, Prediction, time variance, and classification of hydraulic response to recharge in two karst aquifers: Hydrology and Earth System Sciences, v. 17, no. 1, p. 281–294.

Macpherson, G. L., 1996, Hydrogeology of thin limestones: the Konza Prairie Long-Term Ecological Research Site, Northeastern Kansas: Journal of Hydrology, v. 186, no. 1, p. 191–228.

Macpherson, G. L., Roberts, J. A., Blair, J. M., Townsend, M. A., Fowle, D. A., and Beisner, K. R., 2008, Increasing shallow groundwater CO_2 and limestone weathering, Konza Prairie, USA: Geochimica et Cosmochimica Acta, v. 72, no. 23, p. 5,581–5,599.

McKinney, W., 2010, Data structures for statistical computing in Python. In Proceedings of the 9th Python in Science Conference, Austin, Texas, v. 445, no. 1, p. 51–56.

Messager, M. L., Lehner, B., Cockburn, C., Lamouroux, N., Pella, H., Snelder, T., Tockner, K., Trautmann, T., Watt, C., and Datry, T., 2021, Global prevalence of non-perennial rivers and streams: Nature, v. 594, no. 7,863, p. 391–397.

Moore, T., 2023, AET01 Konza prairie grass reference evapotranspiration: Environmental Data Initiative. http://dx.doi.org/10.6073/pasta/b5a57131dc37d035b7c14ec7a8a49604

Nadeau, T.-L., and Rains, M. C., 2007, Hydrological connectivity of headwaters to downstream waters: Introduction to the featured collection: JAWRA Journal of the American Water Resources Association, v. 43, no. 1, p. 1–4.

Nippert, J., 2023, APT01 Daily precipitation amounts measured at multiple sites across Konza Prairie: Environmental Data Initiative.

http://dx.doi.org/10.6073/pasta/434ea64024920232e8260f42f00c3fb6

Nippert, J. B., and Knapp, A. K., 2007, Linking water uptake with rooting patterns in grassland species: Oecologia, v. 153, no. 2, p. 261–272.

Obergfell, C., Bakker, M., and Maas, K., 2019, Estimation of average diffuse aquifer recharge using time series modeling of groundwater heads: Water Resources Research, v. 55, no. 3, p. 2,194–2,210.

Olden, J. D., and Poff, N. L., 2003, Redundancy and the choice of hydrologic indices for characterizing streamflow regimes: River Research and Applications, v. 19, no. 2, p. 101–121.

Price, A. N., Jones, C. N., Hammond, J. C., Zimmer, M. A., and Zipper, S. C., 2021, The drying regimes of non-perennial rivers and streams: Geophysical Research Letters, v. 48, no. 14, e2021GL093298.

Rinderer, M., van Meerveld, H. J., and McGlynn, B. L., 2019, From points to patterns: Using groundwater time series clustering to investigate subsurface hydrological connectivity and runoff source area dynamics: Water Resources Research, v. 55, no. 7, p. 5,784–5,806.

Sadayappan, K., Keen, R., Jarecke, K. M., Moreno, V., Nippert, J. B., Kirk, M. F., Sullivan, P. L., and Li, L., 2023, Drier streams despite a wetter climate in woody-encroached grasslands: Journal of Hydrology, v. 627, 130388.

Samson, F., and Knopf, F., 1994, Prairie conservation in North America: BioScience, v. 44, no. 6, p. 418–421.

Sauquet, E., Shanafield, M., Hammond, J. C., Sefton, C., Leigh, C., and Datry, T., 2021, Classification and trends in intermittent river flow regimes in Australia, northwestern Europe and USA: A global perspective: Journal of Hydrology, v. 597, 126170.

Seybold, E. C., Bergstrom, A., Jones, C. N., Burgin, A. J., Zipper, S., Godsey, S. E., Dodds, W. K., Zimmer, M. A., Shanafield, M., Datry, T., Mazor, R. D., Messager, M. L., Olden, J. D., Ward, A., Yu, S., Kaiser, K. E., Shogren, A., Walker, R. H., 2023, How low can you go? Widespread challenges in measuring low stream discharge and a path forward: Limnology and Oceanography Letters, v. 8, no. 6, p. 804–811.

Shanafield, M., and Cook, P. G., 2014, Transmission losses, infiltration and groundwater recharge through ephemeral and intermittent streambeds: A review of applied methods: Journal of Hydrology, v. 511, p. 518–529.

Shanafield, M., Gutiérrrez-Jurado, K., White, N., Hatch, M., and Keane, R., 2020, Catchment-scale characterization of intermittent stream infiltration: A geophysics approach: Journal of Geophysical Research: Earth Surface, v. 125, no. 2, e2019JF005330.

Shapoori, V., Peterson, T., Western, A., and Costelloe, J., 2015, Decomposing groundwater head variations into meteorological and pumping components: A synthetic study: Hydrogeology Journal, v. 23, no. 7, p. 1431.

Shook, K., and Pomeroy, J., 2012, Changes in the hydrological character of rainfall on the Canadian prairies: Hydrological Processes, v. 26, no. 12, p. 1,752–1,766.

Sullivan, P. L., Stops, M. W., Macpherson, G. L., Li, L., Hirmas, D. R., and Dodds, W. K., 2019, How landscape heterogeneity governs stream water concentration-discharge behavior in carbonate terrains (Konza Prairie, USA): Chemical Geology, v. 527, 118989.

Sullivan, P. L., Zhang, C., Behm, M., Zhang, F., and Macpherson, G. L., 2020, Toward a new conceptual model for groundwater flow in merokarst systems: Insights from multiple geophysical approaches: Hydrological Processes, v. 34, no. 24, p. 4,697–4,711.

Swenson, L. J., Zipper, S., Peterson, D. M., Jones, C. N., Burgin, A. J., Seybold, E., Kirk, M. F., and Hatley, C., 2024, Changes in water age during dry-down of a non-perennial stream: Water Resources Research, v. 60, no. 1, e2023WR034623.

Tramblay, Y., Rutkowska, A., Sauquet, E., Sefton, C., Laaha, G., Osuch, M., Albuquerque, T., Alves, M. H., Banasik, K., Beaufort, A., Brocca, L., Camici, S., Csabai, Z., Dakhlaoui, H., DeGirolamo, A. M., Dörflinger, G., Gallart, F., Gauster, T., Hanich, L. … Kohnová , S., 2021, Trends in flow intermittence for European rivers: Hydrological Sciences Journal, v. 66, no. 1, p. 37–49.

Tsypin, M., and Macpherson, G. L., 2012, The effect of precipitation events on inorganic carbon in soil and shallow groundwater, Konza Prairie LTER Site, NE Kansas, USA: Applied Geochemistry, v. 27, no. 12, p. 2,356–2,369.

Veach, A. M., Dodds, W. K., and Skibbe, A., 2015, Correction: Fire and grazing influences on rates of riparian woody plant expansion along grassland streams: PLOS ONE, v. 10, no. 5, e0129409.

von Asmuth, J. R., and Bierkens, M. F. P., 2005, Modeling irregularly spaced residual series as a continuous stochastic process: Water Resources Research, v. 41, no. 12.

von Asmuth, J. R., Bierkens, M. F. P., and Maas, K., 2002, Transfer function-noise modeling in continuous time using predefined impulse response functions: Water Resources Research, v. 38, no. 12, p. 23-1–23-12.

von Asmuth, J. R., Maas, K., Bakker, M., and Petersen, J., 2008, Modeling time series of ground water head fluctuations subjected to multiple stresses: Groundwater, v. 46, no. 1, p. 30–40.

von Asmuth, J. R., Maas, K., Knotters, M., Bierkens, M. F. P., Bakker, M., Olsthoorn, T. N., Cirkel, D. G., Leunk, I., Schaars, F., and von Asmuth, D. C., 2012, Software for hydrogeologic time series analysis, interfacing data with physical insight: Environmental Modelling & Software, v. 38, p. 178–190.

Warix, S. R., Godsey, S. E., Lohse, K. A., and Hale, R. L., 2021, Influence of groundwater and topography on stream drying in semi-arid headwater streams: Hydrological Processes, v. 35, no. 5, e14185.

Zipper, S., Dallemagne, T., Gleeson, T., Boerman, T. C., and Hartmann, A., 2018, Groundwater pumping impacts on real stream networks: Testing the performance of simple management tools: Water Resources Research, v. 54, no. 8, p. 5,471–5,486.

Zipper, S., Gleeson, T., Li, Q., and Kerr, B., 2021, Comparing streamflow depletion estimation approaches in a heavily stressed, conjunctively managed aquifer: Water Resources

Research, v. 57, no. 2, e2020WR027591.

Zipper, S., Hammond, J. C., Shanafield, M., Zimmer, M., Datry, T., Jones, C. N., Kaiser, K. E., Godsey, S. E., Burrows, R. M., Blaszczak, J. R., Busch, M. H., Price, A. N., Boersma, K. S., Ward, A. S., Costigan, K., Allen, G. H, Krabbenhoft, C. A., Dodds, W. K., Mims, M. C. … Allen, D. C, 2021, Pervasive changes in stream intermittency across the United States: Environmental Research Letters, v. 16, no. 8, 084033.

Zipper, S., Popescu, I., Compare, K., Zhang, C., and Seybold, E. C., 2022, Alternative stable states and hydrological regime shifts in a large intermittent river: Environmental Research Letters, v. 17, no. 7, 074005.

Zipper, S. C., Farmer, W. H., Brookfield, A., Ajami, H., Reeves, H. W., Wardropper, C., Hammond, J. C., Gleeson, T., and Deines, J. M., 2022, Quantifying streamflow depletion from groundwater pumping: A practical review of past and emerging approaches for water management: JAWRA Journal of the American Water Resources Association, v. 58, no. 2, p. 289–312.

Zipper, S. C., Gleeson, T., Kerr, B., Howard, J. K., Rohde, M. M., Carah, J., and Zimmerman, J., 2019, Rapid and accurate estimates of streamflow depletion caused by groundwater pumping using analytical depletion functions: Water Resources Research, v. 55, no. 7, p. 5,807–5,829.

Appendix 1:

Code examples for best EIS and MOR models

In [1]:

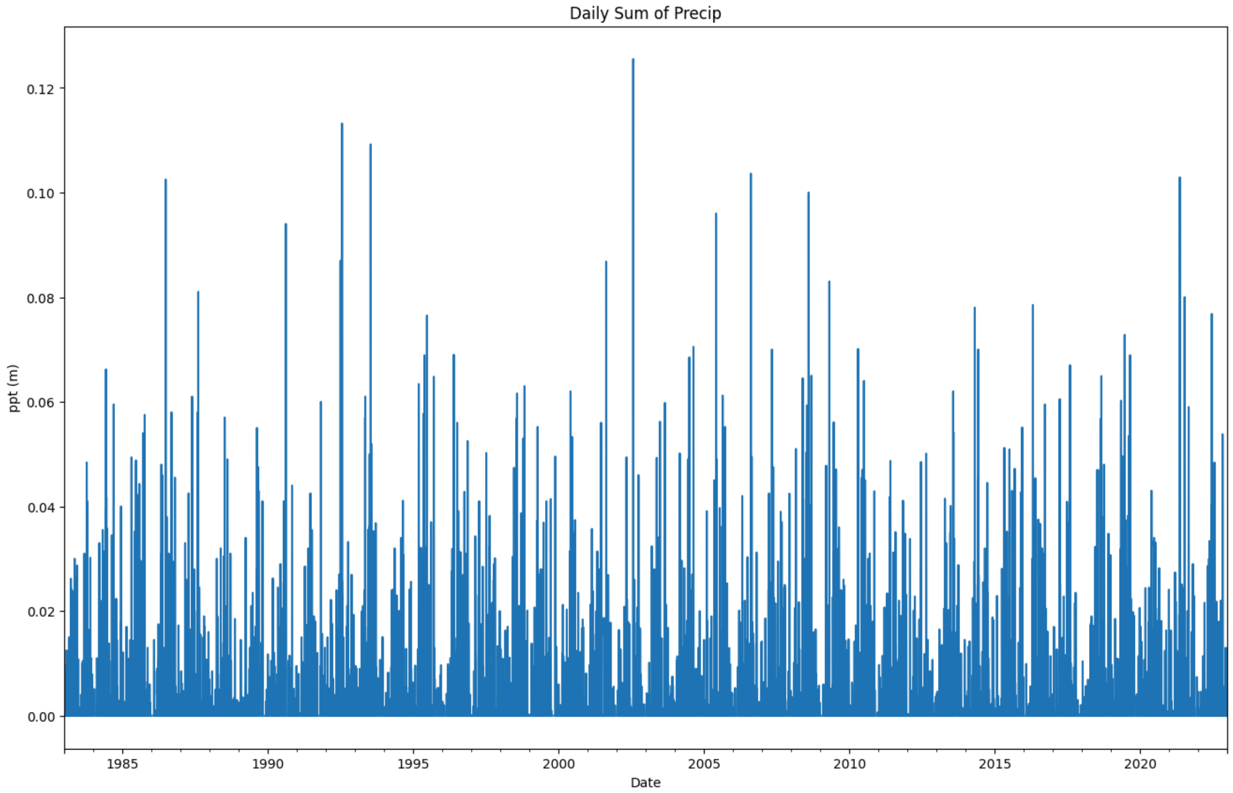
```
# Setup
import numpy as np
import pandas as pd

%matplotlib inline
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (16, 10)  # Set the default figure size (width, heigh
import pastas as ps
from scipy.interpolate import interp1d
import pastas.timeseries as ts
```

Precipitation

In [2]:

```
precip = pd.read_csv(r'n4d_precip_filled.csv', index_col=0)

# Rename the index to 'index'
precip.rename_axis('index', inplace=True)

# Convert the 'index' to a pandas DatetimeIndex
precip.index = pd.to_datetime(precip.index)

# Select the 'ppt' column and create a Series with a name
precip_series = pd.Series(precip['ppt_m'], name='Precipitation')

# Now you have a Series 'precip_series'
# print(precip_series)

print(precip_series.isna().sum())


# Fix duplicates
#precip_series = precip_series.groupby(level=0)
#precip_series = precip_series.mean()
# Validate the stress series using pastas
ps.validate_stress(precip_series)

precip_series.plot(label='Original Data')
plt.xlabel('Date')
plt.ylabel('ppt (m)')
plt.title('Daily Sum of Precip')
plt.show()

# Print the Series
print(precip_series)
```

```
0
```



```
index
1983-01-01    0.0
1983-01-02    0.0
1983-01-03    0.0
1983-01-04    0.0
1983-01-05    0.0
             ...
2022-12-27    0.0
2022-12-28    0.0
2022-12-29    0.0
2022-12-30    0.0
2022-12-31    0.0
Name: Precipitation, Length: 14610, dtype: float64
```

Evapotranspiration

In [3]:

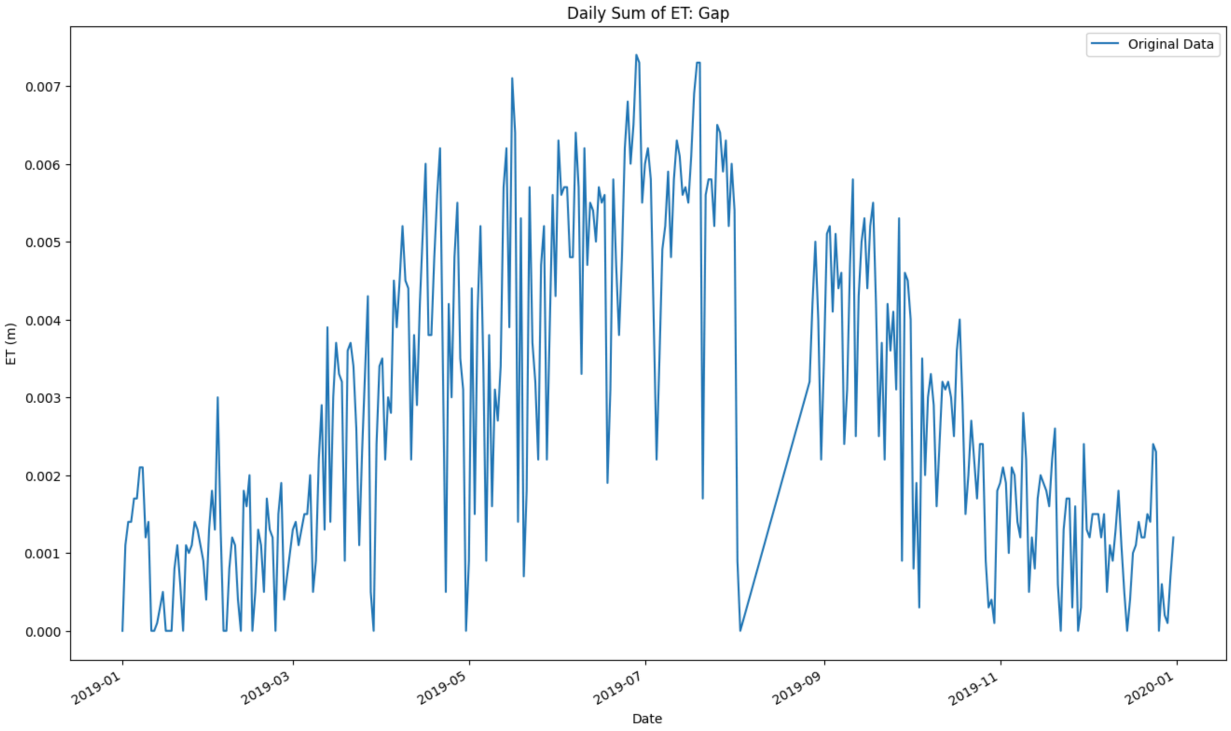
```
# Load your DataFrame from the CSV file
ET_0 = pd.read_csv('AET011.csv')

# Create a new column 'dates' by combining 'recyear', 'recmonth', and 'recday' columns
ET_0['index'] = pd.to_datetime(ET_0['recday'].astype(str) + ' ' + ET_0['recmonth'].ast

# mm to meter
ET_0['DailyET'] = ET_0['DailyET'] / 1000

# Extract the date part (year, month, day) from the 'dates' column
ET_0['index'] = ET_0['index'].dt.date

# Create a new DataFrame with only 'index' and 'DailyET' columns
ET_0 = ET_0[['index', 'DailyET']]

# Convert the 'index' to a pandas DatetimeIndex
ET_0['index'] = pd.to_datetime(ET_0['index'])
```

```
# Create a Series from the 'DailyET' column with a name 'Reference_ET' and set 'index'
ET_0_series = pd.Series(ET_0['DailyET'].values, index=ET_0['index'], name='Reference_E

# Print the Series
print(ET_0_series)
```

```
index
2000-10-16    0.0029
2000-10-17    0.0034
2000-10-18    0.0041
2000-10-19    0.0051
2000-10-20    0.0038
               ...
2022-12-27    0.0010
2022-12-28    0.0010
2022-12-29    0.0020
2022-12-30    0.0010
2022-12-31    0.0012
Name: Reference_ET, Length: 8087, dtype: float64
```

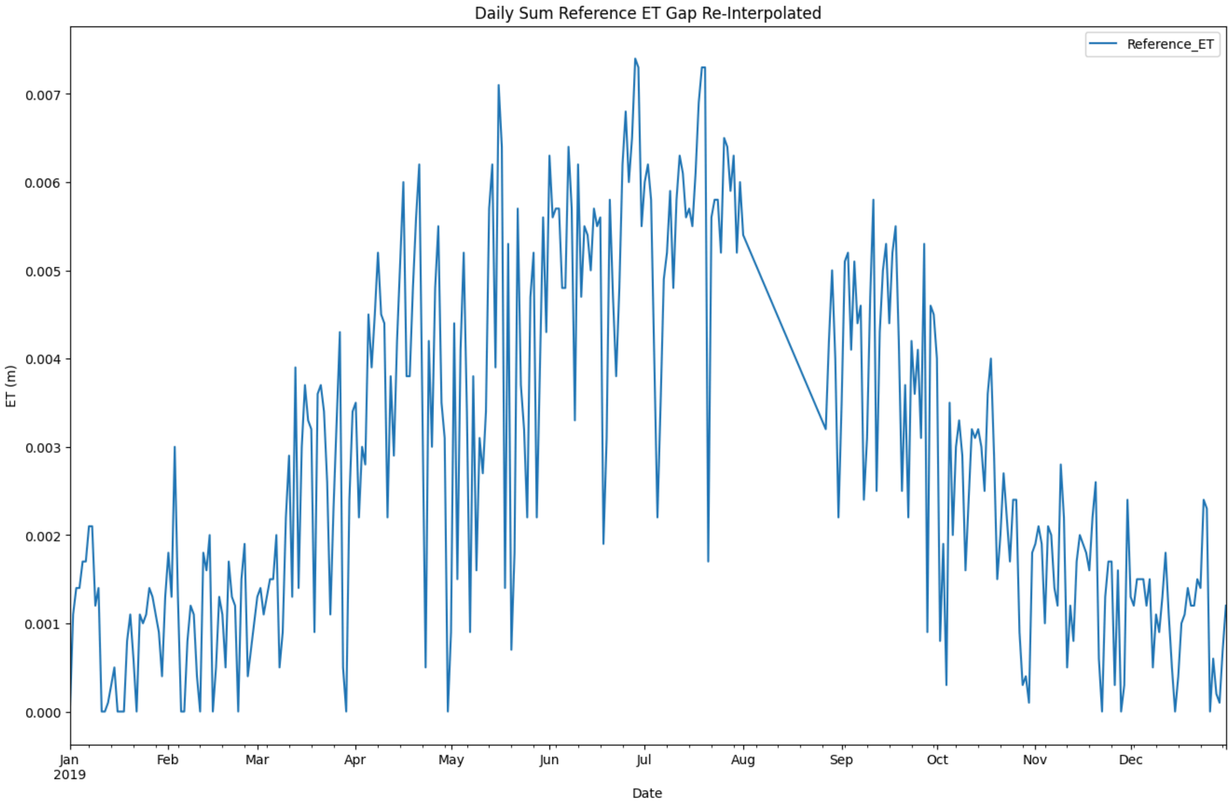
In [4]:

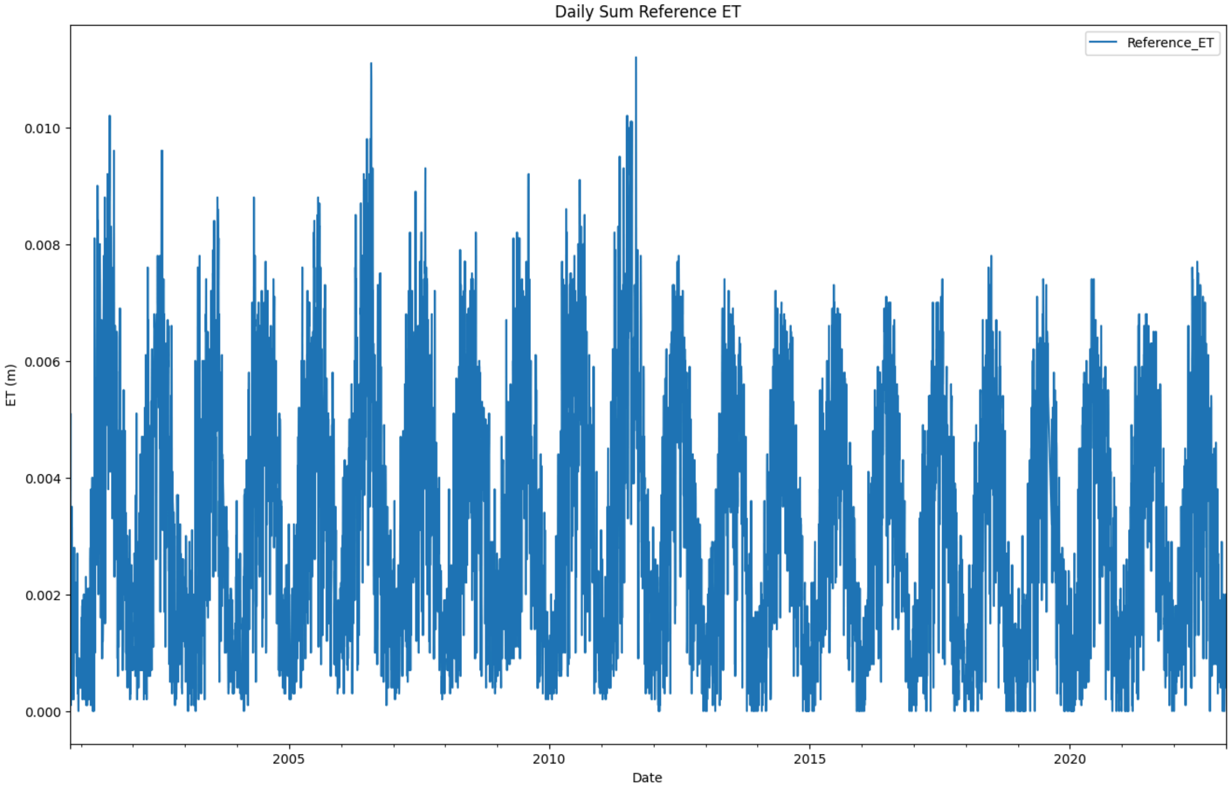
```
# Count the number of missing values (NaN) in the Series
#print(ET_0_series.isna().sum())

# Set a specific date to NA
start_date = '2019-01-01'
end_date = '2019-12-31'
filtered_ET_0_series = ET_0_series[start_date:end_date]

# Plot the filtered data
filtered_ET_0_series.plot(label='Original Data')
plt.xlabel('Date')
plt.ylabel('ET (m)')
plt.title('Daily Sum of ET: Gap')
plt.legend()
plt.show()

target_date = '2019-08-02'
ET_0_series.loc[ET_0_series.index == target_date] = np.nan

target_date = '2019-08-03'
ET_0_series.loc[ET_0_series.index == target_date] = np.nan
```



In [5]:

```
# Fix duplicate time indices
grouped = ET_0_series.groupby(level=0)
ET_0_series = grouped.mean()

# Resample to a regular frequency (e.g., daily)
ET_0_series_resampled = ET_0_series.resample('D').mean()

# Interpolate NAs
ET_0_series_resampled = ET_0_series_resampled.interpolate(method="time")  # For exampl

start_date = '2019-01-01'
end_date = '2019-12-31'
filtered_ET_0_series = ET_0_series_resampled[start_date:end_date]

filtered_ET_0_series.plot()
plt.xlabel('Date')
plt.ylabel('ET (m)')
plt.title('Daily Sum Reference ET Gap Re-Interpolated')
plt.legend()
plt.show()

# Plot all data
plt.figure()
ET_0_series_resampled.plot()
plt.xlabel('Date')
plt.ylabel('ET (m)')
plt.title('Daily Sum Reference ET')
plt.legend()
plt.show()

ps.validate_stress(ET_0_series_resampled)
```





Out[5]:
```
True
```

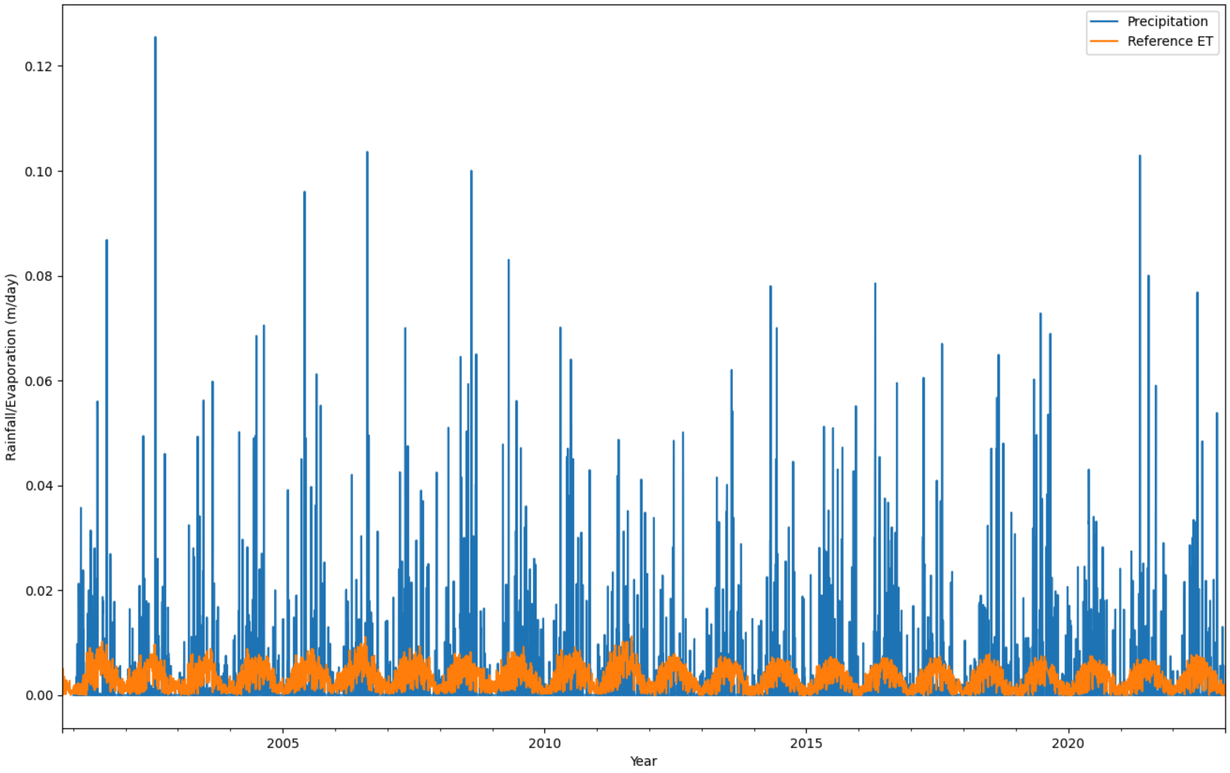
Calculate CWB

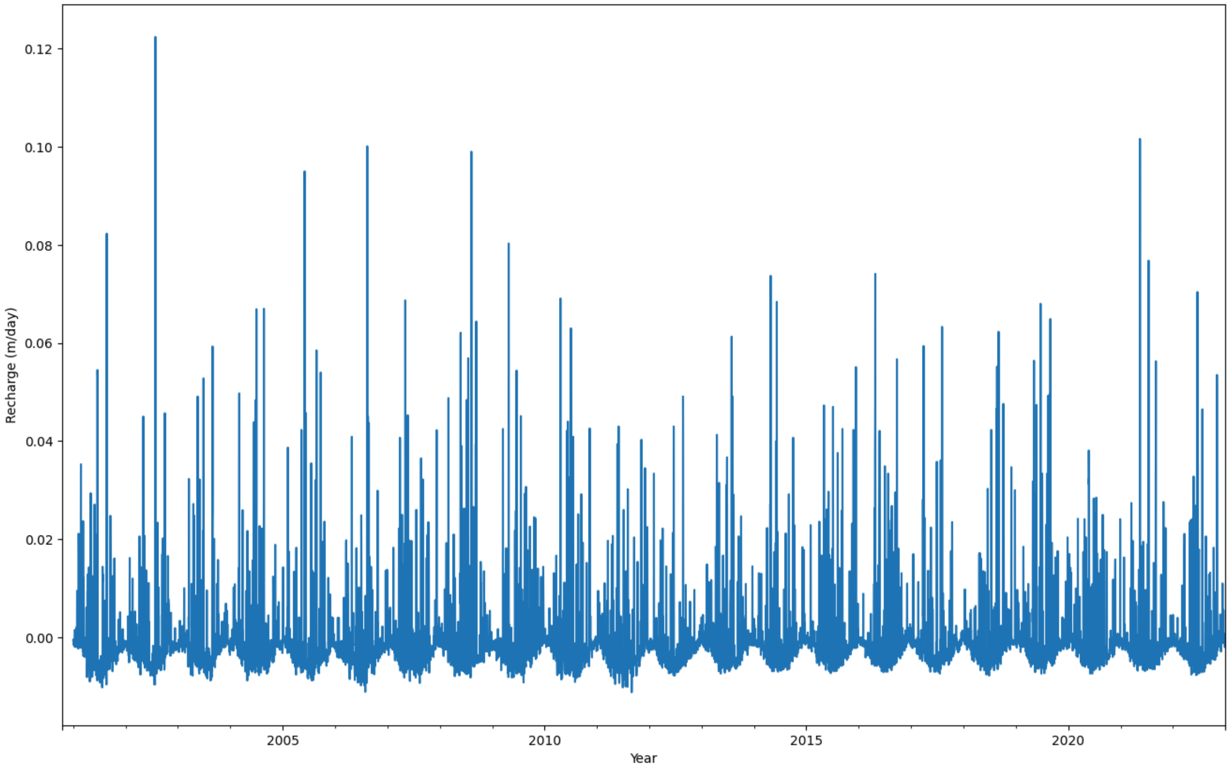
In [6]:
```
# Rename series
evap = ET_0_series_resampled.rename('Evap')
precip = precip_series.rename('Precip')
```

```

# Convert stresses to millimeter
#head = head*1000
evap_mm = evap*1000
precip_mm = precip*1000

# Specify the date range you want to subset
start_date = '2001-01-01'
end_date = '2022-12-31'

# Subset the series by date range using loc
precip = precip.loc[start_date:end_date]
evap - evap.loc[start_date:end_date]

print(precip.isna().sum())
print(evap.isna().sum())

plt.figure()
precip.plot(label="Precipitation")
evap.plot(label="Reference ET")
plt.xlabel("Year")
plt.ylabel("Rainfall/Evaporation (m/day)")
plt.legend(loc="best");

# Calculate the recharge to the groundwater
recharge = precip - evap

recharge.name = "Recharge"  # set name if pandas series
#print("The data type of the recharge series is: %s" % type(recharge))

print(recharge.isna().sum())

# Plot the time series of the precipitation and evaporation
plt.figure()
recharge.plot(label="Recharge")
plt.xlabel("Year")
plt.ylabel("Recharge (m/day)")
```

```
0
0
77
```

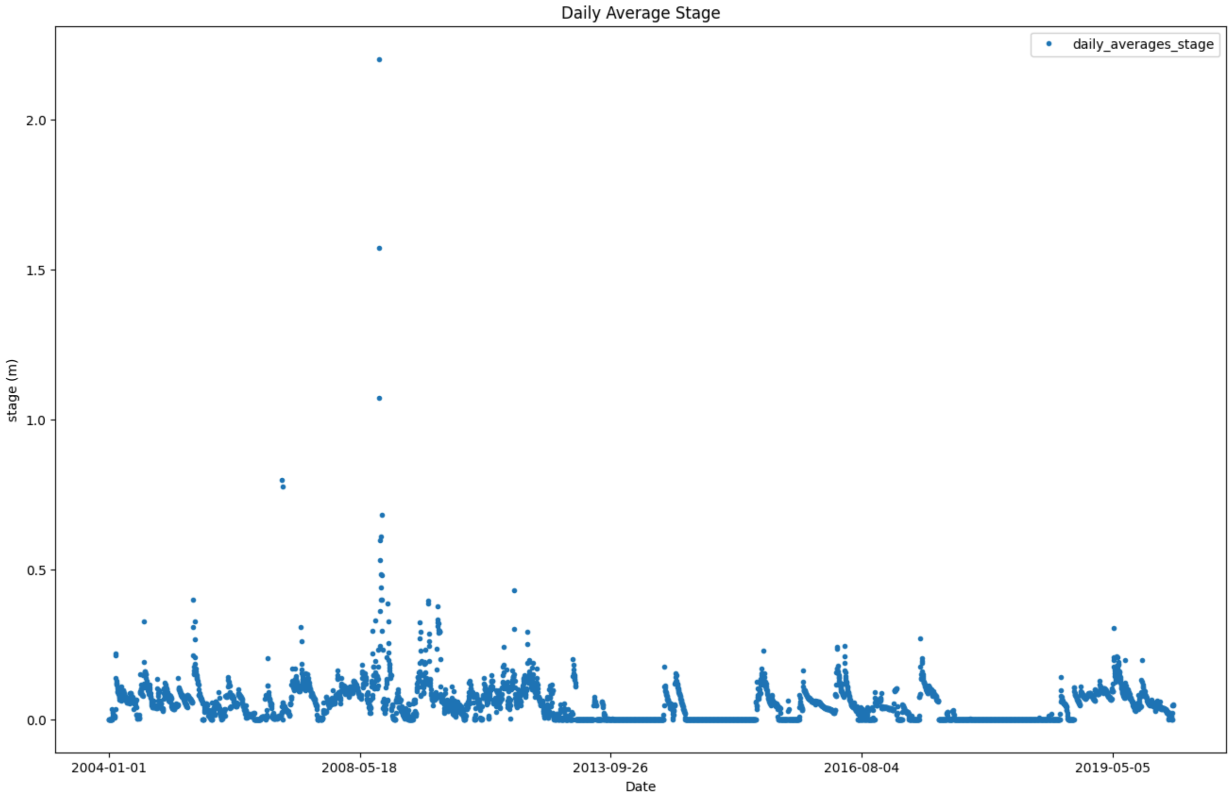
```
Out[6]: Text(0, 0.5, 'Recharge (m/day)')
```



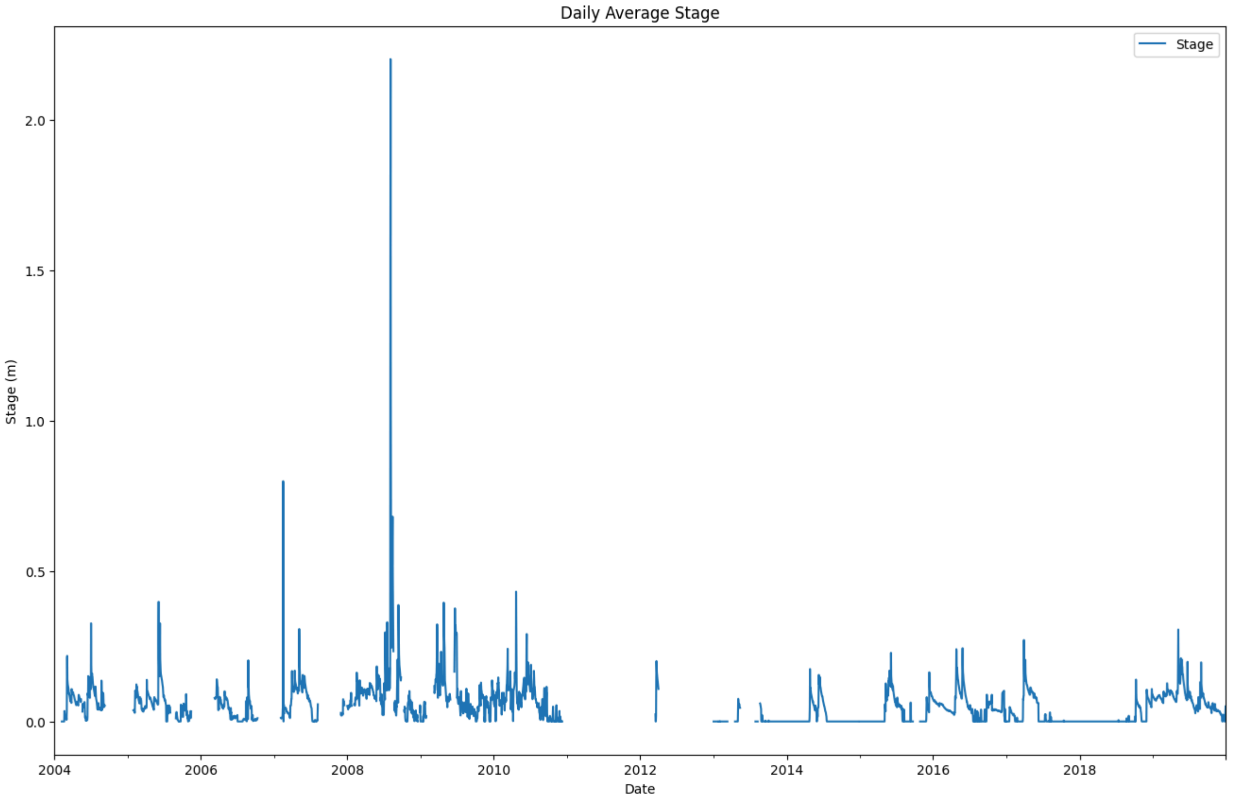


N04D Stage

In [7]:
```
n04d_stage = pd.read_csv(r'ASD021_avg_daily_stage_m.csv', index_col=0)

# Plot data
n04d_stage.plot(style=".",label='Stage')
plt.xlabel('Date')
plt.ylabel('stage (m)')
plt.title('Daily Average Stage')
plt.legend()
plt.show()

# Rename the index to 'index'
n04d_stage.rename_axis('index', inplace=True)

# Convert the 'index' to a pandas DatetimeIndex
n04d_stage.index = pd.to_datetime(n04d_stage.index)

# Select the 'ppt' column and create a Series with a name
waterlevel = pd.Series(n04d_stage['daily_averages_stage'], name='N04D_Stage')


# Reindex the time series to have a complete range of dates
date_range = pd.date_range(start=waterlevel.index.min(), end=waterlevel.index.max(), f
waterlevel = waterlevel.reindex(date_range)

print(waterlevel.isna().sum())

# Plot data
waterlevel.plot(label='Stage')
plt.xlabel('Date')
plt.ylabel('Stage (m)')
plt.title('Daily Average Stage')
plt.legend()
plt.show()
```

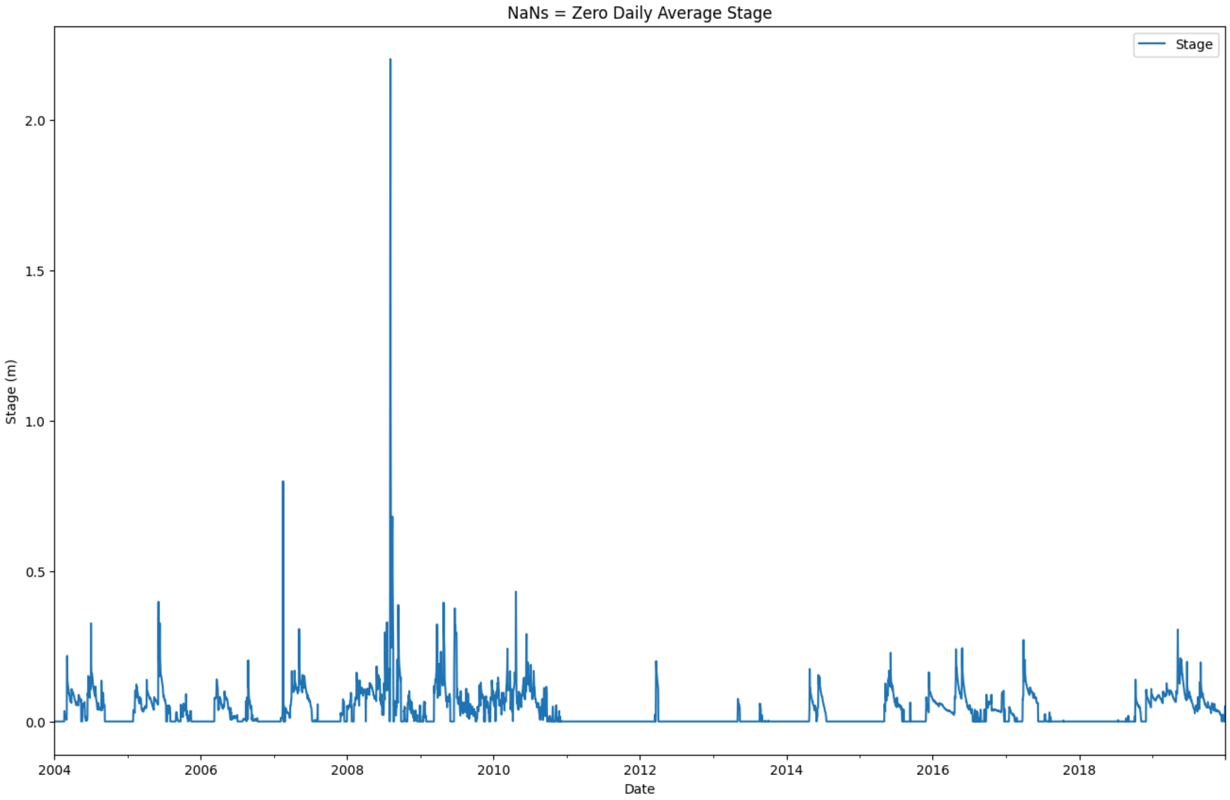


```
1603
```



In [8]:

```
# Fill NAs with 0
waterlevel_zero = waterlevel.fillna(0.0)

# Plot both the original and resampled data
waterlevel_zero.plot(label='Stage')
plt.xlabel('Date')
plt.ylabel('Stage (m)')
plt.title('NaNs = Zero Daily Average Stage')
plt.legend()
plt.show()

# Validate the stress series using pastas
print(ps.validate_stress(waterlevel_zero))

print(waterlevel_zero.isna().sum())
```

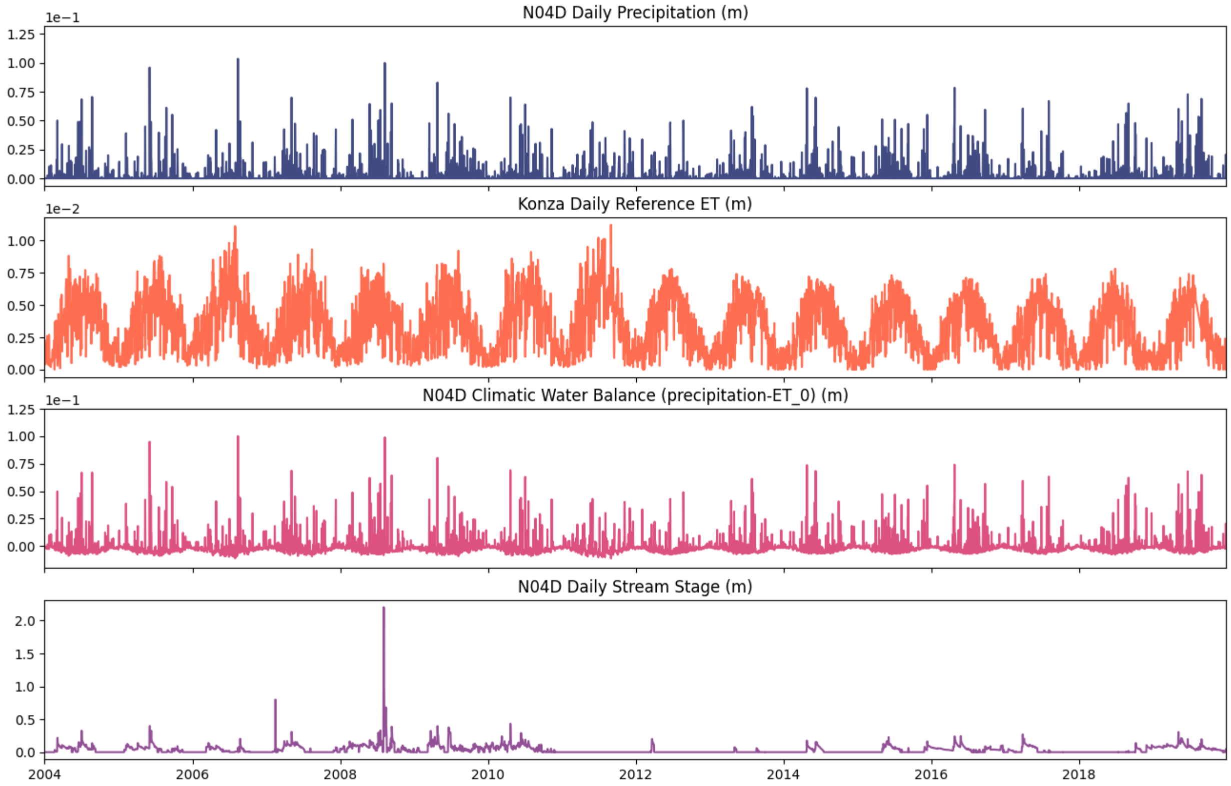


```
True
0
```

All climatic data

```
In [9]: # Create a figure and subplots
fig, (ax1, ax2, ax3, ax4) = plt.subplots(4, 1, sharex=True)

# Plot the data using my_series as the x-values and corresponding y-values
ax1.plot(precip, label='N40D Daily Precipitation', color='#444e86')

ax3.set_ylim(-0.02, 0.125) #Specify limits for the third subplot

# Plot data on each subplot
ax2.plot(evap, label='N40D Reference Evapotranspiration', color='#ff6e54')
ax3.plot(recharge, label='N40D Reference Evapotranspiration', color='#dd5182')
ax4.plot(waterlevel_zero, label='N04D Stream Stage', color='#955196')

# Set the x-axis limits to match the range of my_series
ax4.set_xlim(waterlevel_zero.index.min(), waterlevel_zero.index.max())

# Add titles to subplots
ax1.set_title("N04D Daily Precipitation (m)")
ax2.set_title("Konza Daily Reference ET (m)")
ax4.set_title("N04D Daily Stream Stage (m)")
ax3.set_title("N04D Climatic Water Balance (precipitation-ET_0) (m)")

# Set y-axis of ax2 to scientific notation
ax1.ticklabel_format(style='sci', axis='y', scilimits=(0,0))
ax2.ticklabel_format(style='sci', axis='y', scilimits=(0,0))
ax3.ticklabel_format(style='sci', axis='y', scilimits=(0,0))
```

```
# Show the plot
plt.show()
```

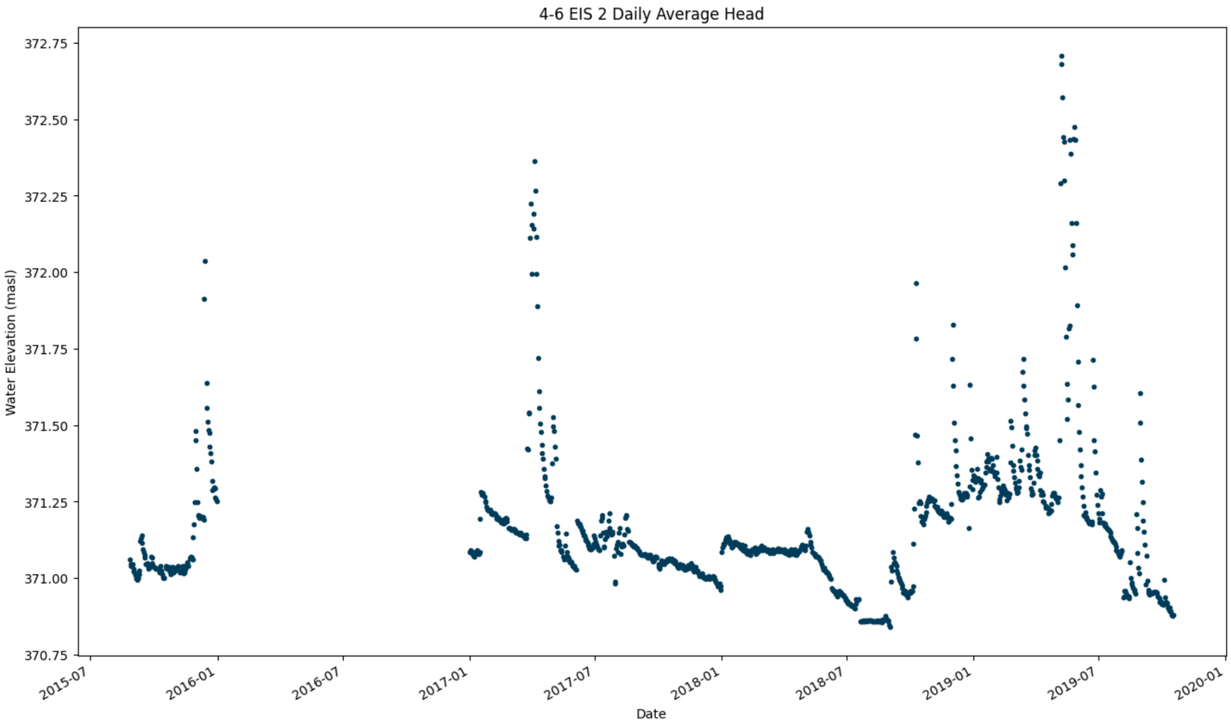


Groundwater Elevation at 4-6 EIS 2: Raw data is in meters

In [10]:

```
EIS2 = pd.read_csv(r'daily_avg_46eis2_df.csv', index_col=0)

# Rename the index to 'index'
EIS2.rename_axis('index', inplace=True)

# Convert the index to datetime format
EIS2.index = pd.to_datetime(EIS2.index)

# Extract the date part and assign it back to the index
EIS2.index = EIS2.index.date

# Create the time series
EIS2_series = pd.Series(EIS2['daily_avg_46eis2'], name='4-6_EIS2_Head', dtype=float)

EIS2_series.index = pd.to_datetime(EIS2_series.index)

# Plot the data
EIS2_series.plot(style='.',label='Water_Elevation', color = '#003f5c')
plt.xlabel('Date')
plt.ylabel('Water Elevation (masl)')
plt.title('4-6 EIS 2 Daily Average Head')

# Save the plot as an SVG file
#plt.savefig('EIS_head.svg', format='svg')

plt.show()
```

```
ps.validate_oseries(EIS2_series)
```



```
WARNING: The Time Series '4-6_EIS2_Head' has nan-values. Pastas will use the fill_nan
settings to fill up the nan-values.
```

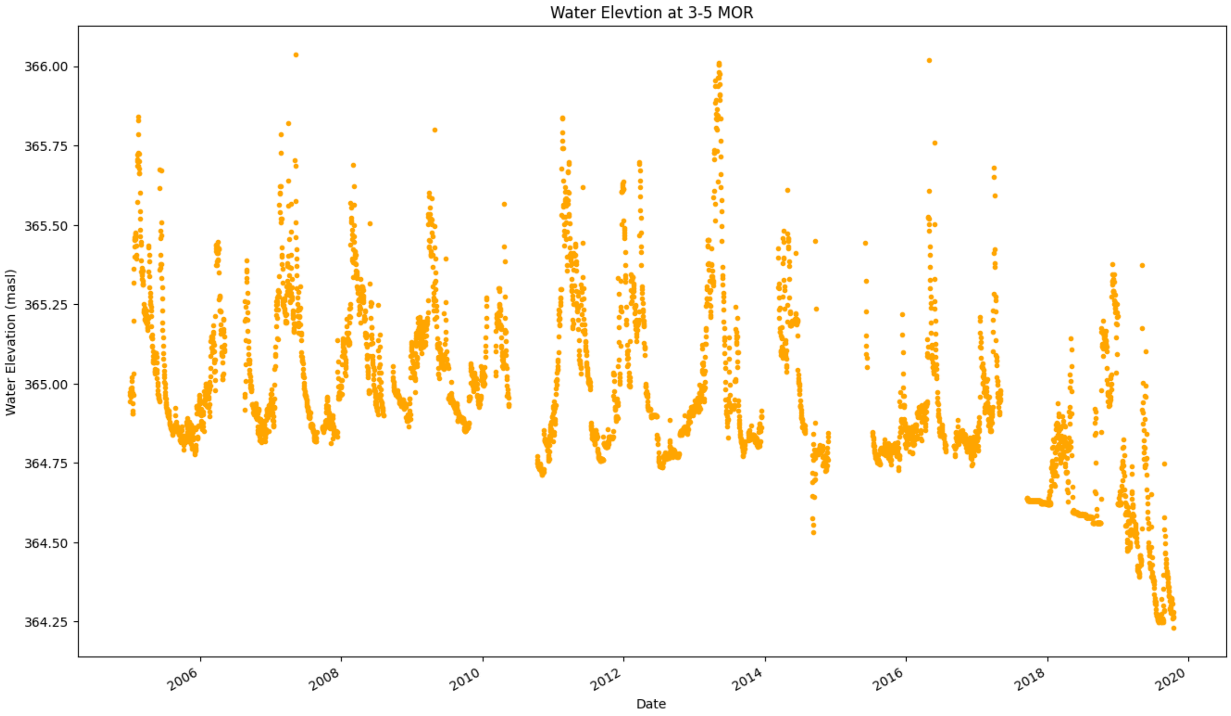
Out[10]:
```
True
```

Groundwater Elevation at 3-5 MOR: Raw data is in meters

In [11]:
```
MOR_3_5 = pd.read_csv(r'daily_avg_35mor_df.csv', index_col=0)

# Rename the index to 'index'
MOR_3_5.rename_axis('index', inplace=True)

# Convert the index to datetime format
MOR_3_5.index = pd.to_datetime(MOR_3_5.index)

# Extract the date part and assign it back to the index
MOR_3_5.index = MOR_3_5.index.date

print(MOR_3_5.head)

#MOR_3_5 = MOR_3_5.set_index('index')['daily_avg_3_5mor']

# Create the MOR35_series Series from the 'daily_avg_3_5mor' column
MOR35_series2 = pd.Series(MOR_3_5['daily_avg_3_5mor'], name='3-5_MOR_Head', dtype=floa

MOR35_series2.index = pd.to_datetime(MOR35_series2.index)

ps.validate_oseries(MOR35_series2)

print(MOR35_series2.head())
```

```
# Plot the data
MOR35_series2.plot(style=".", label='Water_Elevation', color='#ffa600')

plt.xlabel('Date')
plt.ylabel('Water Elevation (masl)')
plt.title('Water Elevtion at 3-5 MOR')
plt.show()
```

```
WARNING: The Time Series '3-5_MOR_Head' has nan-values. Pastas will use the fill_nan
settings to fill up the nan-values.
<bound method NDFrame.head of             daily_avg_3_5mor
2005-01-01        364.972708
2005-01-02        364.943125
2005-01-03        364.963229
2005-01-04        364.985660
2005-01-05        364.984583
...                      ...
2019-12-27               NaN
2019-12-28               NaN
2019-12-29               NaN
2019-12-30               NaN
2019-12-31               NaN

[5074 rows x 1 columns]>
2005-01-01    364.972708
2005-01-02    364.943125
2005-01-03    364.963229
2005-01-04    364.985660
2005-01-05    364.984583
Name: 3-5_MOR_Head, dtype: float64
```

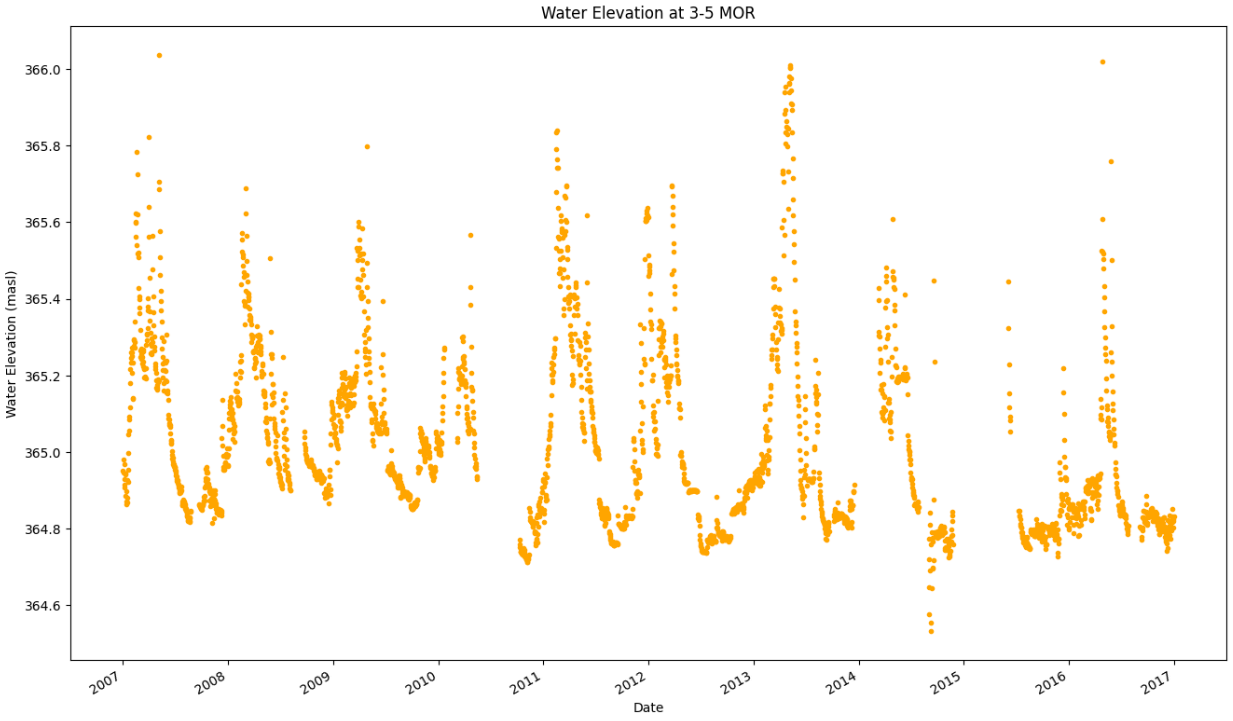


Cut downward trend in MOR data

In [12]:

```
# Specify the date range you want to subset
start_date = '2007'
end_date = '2016'
```

```
# Subset the series by date range using loc
MOR35_series = MOR35_series2.loc[start_date:end_date]

# Plot the data
MOR35_series.plot(style=".", label='Water_Elevation', color='#ffa600')

plt.xlabel('Date')
plt.ylabel('Water Elevation (masl)')
plt.title('Water Elevation at 3-5 MOR')
plt.show()

print(MOR35_series)
```



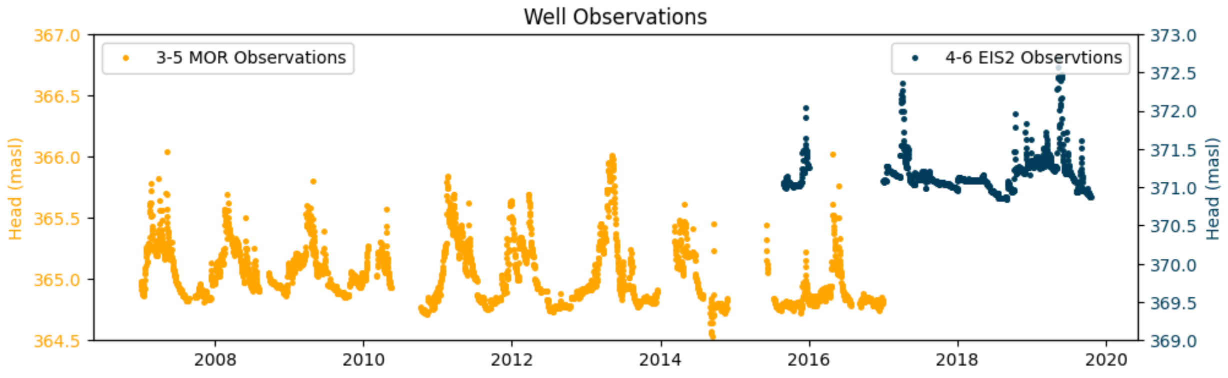
```
2007-01-01    364.951667
2007-01-02    364.945000
2007-01-03    364.981146
2007-01-04           NaN
2007-01-05    364.967986
                 ...
2016-12-27    364.816181
2016-12-28    364.825208
2016-12-29    364.801944
2016-12-30    364.819514
2016-12-31    364.832465
Name: 3-5_MOR_Head, Length: 3249, dtype: float64
```

Both wells

In [13]:

```
# Create a figure and the first set of axes (left y-axis)
fig, ax1 = plt.subplots(figsize=(11, 3.25))

# Plot the second set of data on the right y-axis
ax1.plot(MOR35_series, label='3-5 MOR Observations', color='#ffa600', linestyle='', ma
ax1.set_ylabel('Head (masl)', color='#ffa600')
ax1.tick_params(axis='y', labelcolor='#ffa600')
```

```
ax1.set_ylim(364.5, 367)

# Create a second set of axes (right y-axis) that shares the same x-axis
ax2 = ax1.twinx()

# Set y-limits for the right y-axis
ax2.set_ylim(369, 373)

# Plot the first set of data on the left y-axis
ax2.plot(EIS2_series, label='4-6 EIS2 Observtions', color='#003f5c', linestyle='', mar
#ax1.set_xlabel('Elevation Change (m)')
ax2.set_ylabel('Head (masl)', color='#003f5c')
ax2.tick_params(axis='y', labelcolor='#003f5c')

# Add legends for both sets of data
ax1.legend(loc='upper left')
ax2.legend(loc='upper right')

plt.title('Well Observations')

# Show the plot
plt.show()
```



Best EIS Models

Best Model, Model E7 — 4-6 EIS 2: stage(One) + FlexModel(precip, ET0 (Exp))

In [14]:

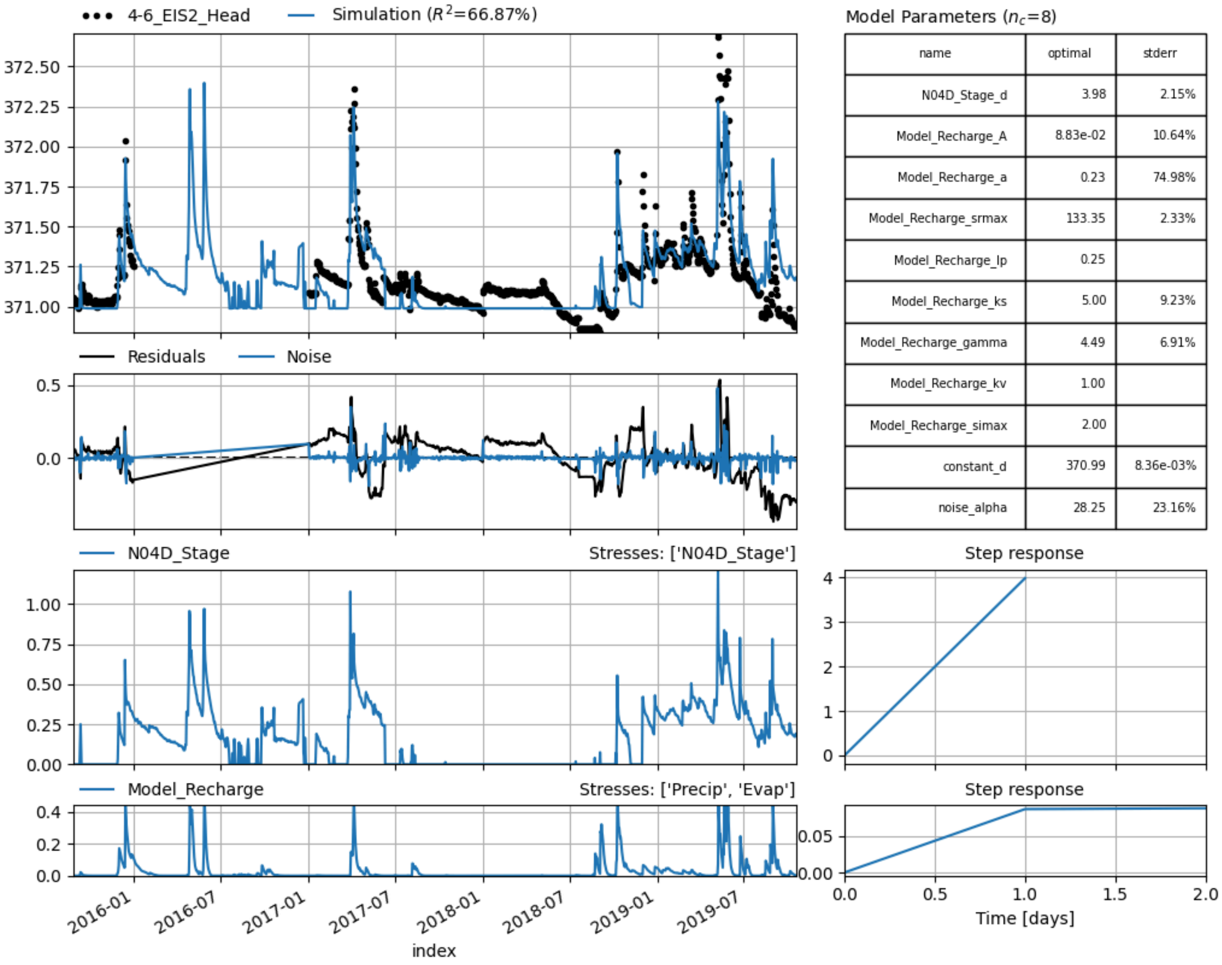
```
ml8b = ps.Model(EIS2_series, name="Non_Linear_Rch_Model")

## ----------------------------------------- Add response function (rfunc) ----------
#rfunc = ps.Exponential() #used for stresses that start to have an immediate effect on
#rfunc =ps.Gamma() #most used response function in Pastas, versatile and able to mimic
#rfunc =ps.Hantush() #used for pumping wells
#rfunc =ps.Polder() #intended for the simulation of the response to variations of surf
#rfunc =ps.FourParam() #four parameters that includes the Exponential, Gamma, Hantush
#rfunc =ps.DoubleExponential() #intended for the simulation in a multi-aquifer system,
#rfunc = ps.Edelman() #describing the propagation of an instantaneous water level chan
rfunc = ps.One() #Instant response with no lag and one parameter d.


sm1 = ps.StressModel(waterlevel_zero, rfunc, name="N04D_Stage", settings=("flux"))
```

```
ml8b.add_stressmodel(sm1)

ml8b.solve(warmup=4015, report=False, noise=False)

ml8b.solve(warmup=4015,initial=False, noise=True, report=False)

#ml3.plots.results()

## ----------------------------------------- Stress model settings -------------------

# up=False #Positive values in stress DO NOT increase simulation values

# Adding a recharge model

## ----------------------------------------- Select a recharge model -----------------
rch = ps.rch.FlexModel()
#rch = ps.rch.Berendrecht()
#rch = ps.rch.Linear()
#rch = ps.rch.Peterson()

## ----------------------------------------- Add response function (rfunc) -----------
rfunc = ps.Exponential() #used for stresses that start to have an immediate effect on
#rfunc =ps.Gamma() #most used response function in Pastas, versatile and able to mimic
#rfunc =ps.Hantush() #used for pumping wells
#rfunc =ps.Polder() #intended for the simulation of the response to variations of surf
#rfunc =ps.FourParam() #four parameters that includes the Exponential, Gamma, Hantush
#rfunc =ps.DoubleExponential() #intended for the simulation in a multi-aquifer system,
#rfunc = ps.Edelman() #describing the propagation of an instantaneous water level chan
#rfunc = ps.One() #Instant response with no lag and one parameter d.

sm3 = ps.RechargeModel(precip_mm, evap_mm, rfunc = rfunc, name="Model_Recharge", recha

ml8b.add_stressmodel(sm3)

# ## ----------------------------------------- Setting parameters --------------------
# param = "rainevap_srmax"
# # options: initial = 15, vary = True or False
# ml2.set_parameter(param, vary=False) #initial=0.300
# ml2.set_parameter("rainevap_gf", vary=False)
# ml2.set_parameter("rainevap_alpha", initial=0.95)
## ----------------------------------------- Solve settings --------------------------
# noise=False #solve without noise model
# report=False
# initial=False #use optimized parameters from previous solve
# Example: First: ml.solve(noise=False, report=False)
#          Second: ml.solve(noise=True, initial=False)
ml8b.solve(warmup=4015,noise=False,  report=False) #, tmin=begin_date, tmax=end_date,

ml8b.solve(warmup=4015, noise=True, initial=False)

ml8b.plots.results()

# Calculate KGE
ml8b.stats.kge_2012()
```

```
WARNING: The Time Series '4-6_EIS2_Head' has nan-values. Pastas will use the fill_nan
settings to fill up the nan-values.
INFO: Time Series '4-6_EIS2_Head': 110 nan-value(s) was/were found and filled with: d
rop.
```

```
Fit report Non_Linear_Rch_Model                Fit Statistics
================================================================
nfev     16                        EVP                     66.94
nobs     1112                      R2                       0.67
noise    True                      RMSE                     0.14
tmin     2015-08-27 00:00:00       AIC                  -7250.49
tmax     2019-10-18 00:00:00       BIC                  -7210.38
freq     D                         Obj                      0.81
warmup   4015 days 00:00:00        ___                          
solver   LeastSquares              Interp.                    No

Parameters (8 optimized)
================================================================
                             optimal      stderr     initial   vary
N04D_Stage_d                3.984553      ±2.15%    3.192915   True
Model_Recharge_A            0.088283     ±10.64%    0.137455   True
Model_Recharge_a            0.231712     ±74.98%    0.712608   True
Model_Recharge_srmax      133.354144      ±2.33%  152.352702   True
Model_Recharge_lp           0.250000        ±nan    0.250000  False
Model_Recharge_ks           5.000792      ±9.23%    5.689597   True
Model_Recharge_gamma        4.486948      ±6.91%   11.148583   True
Model_Recharge_kv           1.000000        ±nan    1.000000  False
Model_Recharge_simax        2.000000        ±nan    2.000000  False
constant_d                370.989152  ±8.36e-03%  371.027261   True
noise_alpha                28.249013     ±23.16%    1.000000   True
```

Out[14]:
```
0.8185248407240155
```



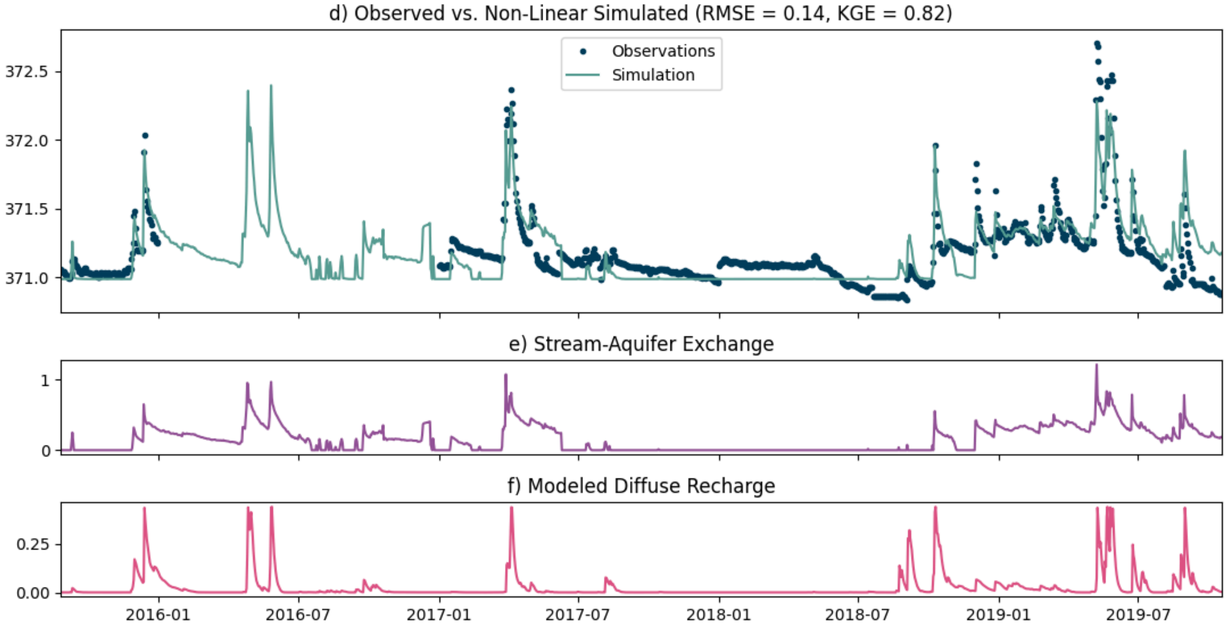
Model Parameters (n_c=8)

name	optimal	stderr
N04D_Stage_d	3.98	2.15%
Model_Recharge_A	8.83e-02	10.64%
Model_Recharge_a	0.23	74.98%
Model_Recharge_srmax	133.35	2.33%
Model_Recharge_lp	0.25	
Model_Recharge_ks	5.00	9.23%
Model_Recharge_gamma	4.49	6.91%
Model_Recharge_kv	1.00	
Model_Recharge_simax	2.00	
constant_d	370.99	8.36e-03%
noise_alpha	28.25	23.16%

In [15]:

```
# Get the observed and simulated time series
contributions = ml8b.get_contributions()
print(contributions)

observed_8b = ml8b.oseries.series
simulated_8b = ml8b.simulate()

stage_contr = contributions[0]
modeled_recharge = contributions[1]

# Create a figure and subplots with different heights
fig, (ax1, ax2, ax3) = plt.subplots(3, 1, sharex=True, gridspec_kw={'height_ratios': [

# Plot the observed and simulated time series on ax1
ax1.plot(observed_8b.index, observed_8b, label="Observations", color='#003f5c', linest
ax1.plot(simulated_8b.index, simulated_8b, label="Simulation", color="#599e94")

# Set the x-axis limits to match the range of my_series
ax1.set_xlim(observed_8b.index.min(), observed_8b.index.max())

# Plot data on each subplot
ax2.plot(stage_contr, label='Stream-Aquifer Exchange', color='#955196')
ax3.plot(modeled_recharge, label='Diffuse Recharge', color='#dd5182')

# Add titles to subplots
ax1.set_title("d) Observed vs. Non-Linear Simulated (RMSE = 0.14, KGE = 0.82)")
ax2.set_title("e) Stream-Aquifer Exchange")
ax3.set_title("f) Modeled Diffuse Recharge")

ax1.legend(loc='upper center')

# Show the plot
plt.tight_layout()
plt.show()
```

```
[2015-08-27    0.000000
2015-08-28    0.000000
2015-08-29    0.000000
2015-08-30    0.000000
2015-08-31    0.000000
                ...
2019-10-14    0.175764
2019-10-15    0.172718
2019-10-16    0.171574
2019-10-17    0.185048
2019-10-18    0.190067
Freq: D, Name: N04D_Stage, Length: 1514, dtype: float64, index
2015-08-27    0.000026
2015-08-28    0.000012
2015-08-29    0.000086
2015-08-30    0.000087
2015-08-31    0.000066
                ...
2019-10-14    0.007019
2019-10-15    0.005271
2019-10-16    0.003962
2019-10-17    0.003078
2019-10-18    0.002097
Name: Model_Recharge, Length: 1514, dtype: float64]
```



2nd Best Model: Model E4 — 4-6 EIS 2: stage(One) + CWB(FourP)

In [16]:

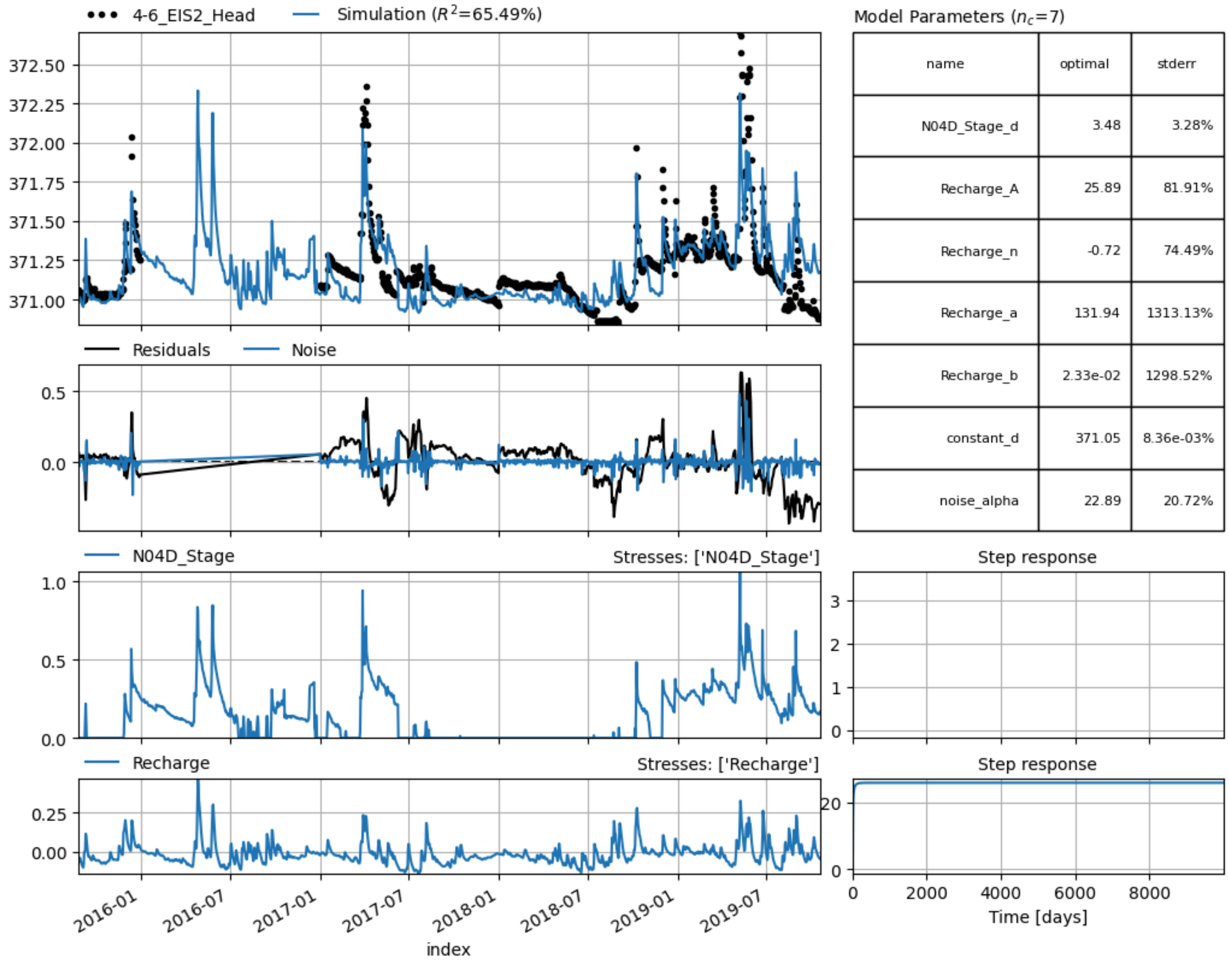
```
ml4a = ps.Model(EIS2_series, name="4-6_EIS2_Head_FourParam")

## ----------------------------------------- Add response function (rfunc) ----------
#rfunc = ps.Exponential() #used for stresses that start to have an immediate effect on
#rfunc =ps.Gamma() #most used response function in Pastas, versatile and able to mimic
#rfunc =ps.Hantush() #used for pumping wells
#rfunc =ps.Polder() #intended for the simulation of the response to variations of surf
#rfunc =ps.FourParam() #four parameters that includes the Exponential, Gamma, Hantush
#rfunc =ps.DoubleExponential() #intended for the simulation in a multi-aquifer system,
#rfunc = ps.Edelman() #describing the propagation of an instantaneous water level chan
rfunc = ps.One() #Instant response with no lag and one parameter d.


sm1 = ps.StressModel(waterlevel_zero, rfunc, name="N04D_Stage", settings=("flux"))

ml4a.add_stressmodel(sm1)

ml4a.solve(warmup = 4015,report=False, noise=False)#warmup = warmup, tmin=begin_date,

#ml1.set_parameter("recharge_n", vary=False)

ml4a.solve(warmup = 4015,initial=False, noise=True, report=False)

#ml4.plots.results()

## ----------------------------------------- Add response function (rfunc) ----------
#rfunc = ps.Exponential() #used for stresses that start to have an immediate effect on
#rfunc =ps.Gamma() #most used response function in Pastas, versatile and able to mimic
#rfunc =ps.Hantush() #used for pumping wells
#rfunc =ps.Polder() #intended for the simulation of the response to variations of surf
rfunc =ps.FourParam() #four parameters that includes the Exponential, Gamma, Hantush a
#rfunc =ps.DoubleExponential() #intended for the simulation in a multi-aquifer system,
#rfunc = ps.Edelman() #describing the propagation of an instantaneous water level chan
#rfunc = ps.One() #Instant response with no lag and one parameter d.
```

```python
sm2 = ps.StressModel(recharge, rfunc, name="Recharge", settings=("evap"))

ml4a.add_stressmodel(sm2)

ml4a.solve(warmup = 4015,report=False, noise=False)

#ml1.set_parameter("recharge_n", vary=False)

ml4a.solve(warmup = 4015,initial=False, noise=True)

ml4a.plots.results()

ml4a.stats.kge_2012()
```

```
WARNING: The Time Series '4-6_EIS2_Head' has nan-values. Pastas will use the fill_nan
settings to fill up the nan-values.
INFO: Time Series '4-6_EIS2_Head': 110 nan-value(s) was/were found and filled with: d
rop.
WARNING: The Time Series 'Recharge' has nan-values. Pastas will use the fill_nan sett
ings to fill up the nan-values.
INFO: Time Series 'Recharge': 77 nan-value(s) was/were found and filled with: interpo
late.
INFO: Time Series 'Recharge': 77 nan-value(s) was/were found and filled with: interpo
late.
```

```
Fit report 4-6_EIS2_Head_FourParam      Fit Statistics
======================================================
nfev    95                     EVP               65.53
nobs    1112                   R2                 0.65
noise   True                   RMSE               0.14
tmin    2015-08-27 00:00:00    AIC            -6999.55
tmax    2019-10-18 00:00:00    BIC            -6964.45
freq    D                      Obj                1.01
warmup  4015 days 00:00:00     ___
solver  LeastSquares           Interp.              No

Parameters (7 optimized)
======================================================
                 optimal      stderr     initial  vary
N04D_Stage_d    3.475867      ±3.28%    3.455245  True
Recharge_A     25.887989     ±81.91%   19.065919  True
Recharge_n     -0.715955     ±74.49%   -0.020019  True
Recharge_a    131.941459   ±1313.13%    7.653609  True
Recharge_b      0.023294   ±1298.52%    0.324403  True
constant_d    371.051306  ±8.36e-03%  371.052778  True
noise_alpha    22.890860     ±20.72%    1.000000  True

Warnings! (1)
======================================================
Response tmax for 'Recharge' > than calibration period.
```

```
Out[16]: 0.7551121382641586
```

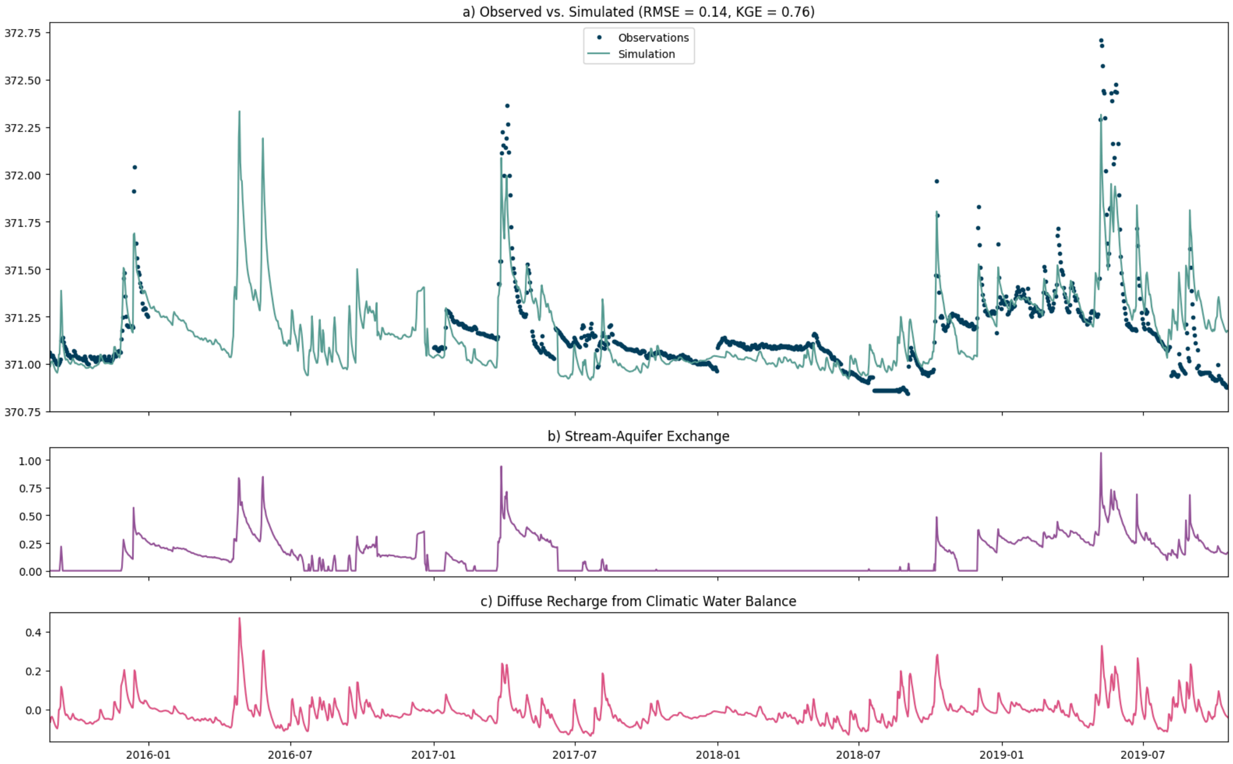


name	optimal	stderr
N04D_Stage_d	3.48	3.28%
Recharge_A	25.89	81.91%
Recharge_n	-0.72	74.49%
Recharge_a	131.94	1313.13%
Recharge_b	2.33e-02	1298.52%
constant_d	371.05	8.36e-03%
noise_alpha	22.89	20.72%

In [17]:

```
# Get the contributions of individual stress models
contributions = ml4a.get_contributions()
print(contributions)

stage_contr_4a = contributions[0]
recharge_contr_4a = contributions[1]

# Get the observed and simulated time series
observed_4a = ml4a.oseries.series
simulated_4a = ml4a.simulate()

# Create a figure and subplots with different heights
fig, (ax1, ax2, ax3) = plt.subplots(3, 1, sharex=True, gridspec_kw={'height_ratios': [

# Plot the data using my_series as the x-values and corresponding y-values
# Plot the observed and simulated time series on ax1
ax1.plot(observed_4a.index, observed_4a, label="Observations", color='#003f5c', linest
ax1.plot(simulated_4a.index, simulated_4a, label="Simulation", color="#599e94")

# Set the x-axis limits to match the range of my_series
ax1.set_xlim(observed_4a.index.min(), observed_4a.index.max())

# Plot data on each subplot
ax2.plot(stage_contr_4a, label='Stream-Aquifer Exchange', color='#955196')
ax3.plot(recharge_contr_4a, label='Diffuse Recharge', color='#dd5182')

# Add titles to subplots
ax1.set_title("a) Observed vs. Simulated (RMSE = 0.14, KGE = 0.76)")
ax2.set_title("b) Stream-Aquifer Exchange")
ax3.set_title("c) Diffuse Recharge from Climatic Water Balance")

ax1.legend(loc='upper center')

# Show the plot
plt.tight_layout()
plt.show()

print(ml4a.stats.kge_2012())
```

```
[2015-08-27    0.000000
2015-08-28    0.000000
2015-08-29    0.000000
2015-08-30    0.000000
2015-08-31    0.000000
                ...
2019-10-14    0.153326
2019-10-15    0.150668
2019-10-16    0.149670
2019-10-17    0.161424
2019-10-18    0.165802
Freq: D, Name: N04D_Stage, Length: 1514, dtype: float64, index
2015-08-27   -0.065916
2015-08-28   -0.066312
2015-08-29   -0.042091
2015-08-30   -0.037240
2015-08-31   -0.043167
                ...
2019-10-14   -0.023074
2019-10-15   -0.030008
2019-10-16   -0.034772
2019-10-17   -0.038084
2019-10-18   -0.042347
Name: Recharge, Length: 1514, dtype: float64]
```



a) Observed vs. Simulated (RMSE = 0.14, KGE = 0.76)

b) Stream-Aquifer Exchange

c) Diffuse Recharge from Climatic Water Balance

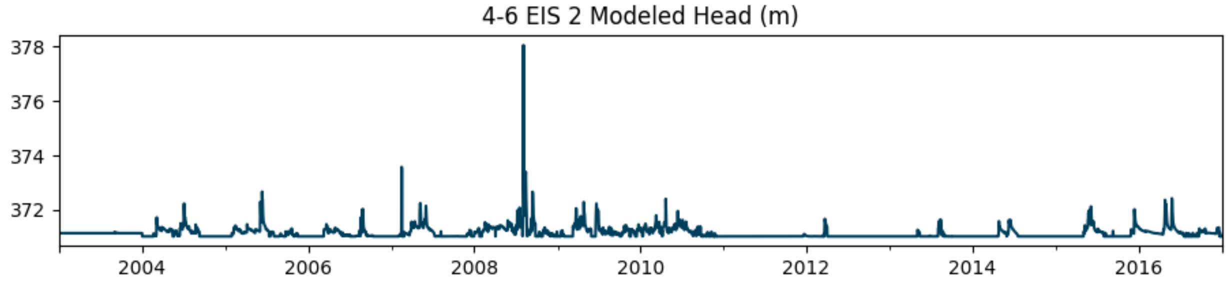
```
0.7551121382641586
```

Extracting EIS simulation

In [18]:

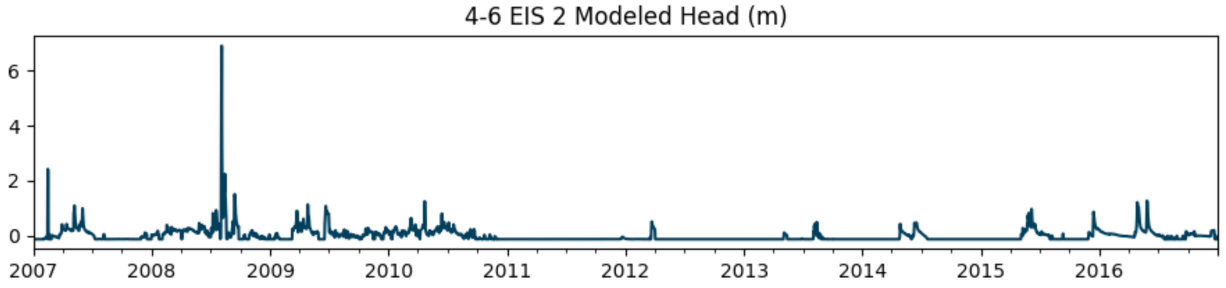
```
p = ml8b.parameters.initial.values

EIS_model = ml8b.simulate(p=p,tmin="2003", tmax="2017");

EIS_model.plot(color='#003f5c',figsize=(11, 2))
plt.title('4-6 EIS 2 Modeled Head (m)')

plt.show()
```

```
INFO: Time Series 'N04D_Stage' was extended in the past to 1992-01-04 00:00:00 with t
he mean value (0.037) of the time series.
INFO: Time Series 'Evap' was extended in the past to 1992-01-04 00:00:00 with the mea
n value (3.3) of the time series.
```



In [19]:

```
EIS_mdl_mean = np.mean(EIS_model)

EIS_model_norm = EIS_model-EIS_mdl_mean

start_date = '2007'
end_date = '2016'

# Subset the series by date range using loc
EIS_model_norm2 = EIS_model_norm.loc[start_date:end_date]

EIS_model_norm2.plot(color='#003f5c',figsize=(11, 2))
plt.title('4-6 EIS 2 Modeled Head (m)')
plt.show()

print(ps.validate_stress(EIS_model_norm))
```



```
True
```

Best MOR Models

2nd Best, Model M7 — 3-5 MOR: rch.Linear(precip, ET0 (DblExp)) + stage(Gam) + EIS simulated head(Gam)

In [20]:
```
begin_date = "2007"
end_date = "2017"
warmup = 1095

smdl = ps.Model(MOR35_series, name="MOR_Special")

# ----------------------------------------- Select a recharge model -----------------
#rch = ps.rch.FlexModel(gw_uptake=True)
#rch = ps.rch.Berendrecht()
rch = ps.rch.Linear()
#rch = ps.rch.Peterson()

## ----------------------------------------- Add response function (rfunc) ----------
#rfunc = ps.Exponential() #used for stresses that start to have an immediate effect on
#rfunc =ps.Gamma() #most used response function in Pastas, versatile and able to mimic
#rfunc =ps.Hantush() #used for pumping wells
#rfunc =ps.Polder() #intended for the simulation of the response to variations of surf
#rfunc =ps.FourParam() #four parameters that includes the Exponential, Gamma, Hantush
rfunc =ps.DoubleExponential() #intended for the simulation in a multi-aquifer system,
#rfunc = ps.Edelman() #describing the propagation of an instantaneous water level chan
#rfunc = ps.One() #Instant response with no lag and one parameter d.

sm1 = ps.RechargeModel(precip_mm, evap_mm, rfunc = rfunc, name="Linear_Model_Recharge"

smdl.add_stressmodel(sm1)

smdl.solve(warmup = warmup, tmin=begin_date, tmax=end_date, noise=False,  report=False

smdl.solve(warmup = warmup, tmin=begin_date, tmax=end_date, noise=True, initial=False,

#ml8.plots.results()

# ----------------------------------------- Add response function (rfunc) -----------
#rfunc = ps.Exponential() #used for stresses that start to have an immediate effect on
rfunc =ps.Gamma() #most used response function in Pastas, versatile and able to mimic
#rfunc =ps.Hantush() #used for pumping wells
#rfunc =ps.Polder() #intended for the simulation of the response to variations of surf
#rfunc =ps.FourParam() #four parameters that includes the Exponential, Gamma, Hantush
#rfunc =ps.DoubleExponential() #intended for the simulation in a multi-aquifer system,
#rfunc = ps.Edelman() #describing the propagation of an instantaneous water level chan
#rfunc = ps.One() #Instant response with no lag and one parameter d.

sm2 = ps.StressModel(waterlevel_zero, rfunc, name="N04D_Stage", settings=("waterlevel'

smdl.add_stressmodel(sm2)

smdl.solve(warmup = warmup, tmin=begin_date, tmax=end_date, report=False, noise=False)

smdl.solve(warmup = warmup, tmin=begin_date, tmax=end_date, initial=False, noise=True,

#smdl.plots.results()

## ----------------------------------------- Add response function (rfunc) ----------
#rfunc = ps.Exponential() #used for stresses that start to have an immediate effect on
```

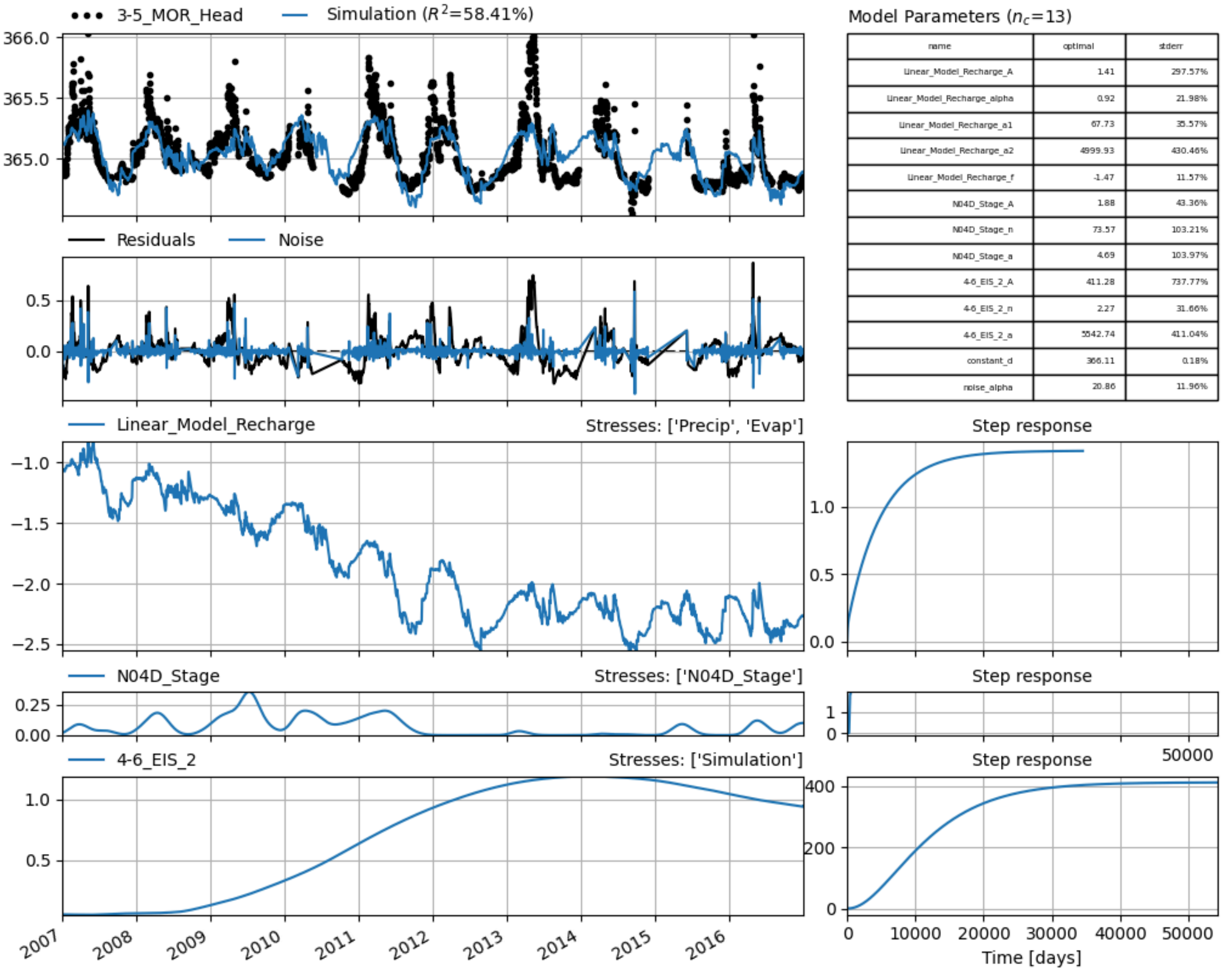
```
rfunc =ps.Gamma() #most used response function in Pastas, versatile and able to mimic
#rfunc =ps.Hantush() #used for pumping wells
#rfunc =ps.Polder() #intended for the simulation of the response to variations of surf
#rfunc =ps.FourParam() #four parameters that includes the Exponential, Gamma, Hantush
#rfunc =ps.DoubleExponential() #intended for the simulation in a multi-aquifer system,
#rfunc = ps.Edelman() #describing the propagation of an instantaneous water level chan
#rfunc = ps.One() #Instant response with no lag and one parameter d.

sm3 = ps.StressModel(EIS_model_norm, rfunc, name="4-6_EIS_2", settings=("prec"))

smdl.add_stressmodel(sm3)


smdl.solve(warmup = warmup, tmin=begin_date, tmax=end_date, report=False, noise=False)

smdl.solve(warmup = warmup, tmin=begin_date, tmax=end_date, initial=False, noise=True)

smdl.plots.results()

smdl.stats.kge_2012()
```

```
WARNING: The Time Series '3-5_MOR_Head' has nan-values. Pastas will use the fill_nan
settings to fill up the nan-values.
INFO: Time Series '3-5_MOR_Head': 340 nan-value(s) was/were found and filled with: dr
op.
```

```
Fit report MOR_Special                                       Fit Statistics
======================================================================
nfev    43                     EVP                                   58.45
nobs    2909                   R2                                     0.58
noise   True                   RMSE                                   0.15
tmin    2007-01-01 00:00:00    AIC                               -17766.87
tmax    2016-12-31 00:00:00    BIC                               -17689.19
freq    D                      Obj                                    3.21
warmup  1095 days 00:00:00     ___
solver  LeastSquares           Interp.                                  No

Parameters (13 optimized)
======================================================================
                                  optimal     stderr       initial   vary
Linear_Model_Recharge_A          1.414700   ±297.57%      0.734205   True
Linear_Model_Recharge_alpha      0.920233    ±21.98%      0.869215   True
Linear_Model_Recharge_a1        67.728865    ±35.57%     80.132643   True
Linear_Model_Recharge_a2      4999.926795   ±430.46%   3758.676057   True
Linear_Model_Recharge_f         -1.473907    ±11.57%     -2.000000   True
N04D_Stage_A                     1.881371    ±43.36%      1.870409   True
N04D_Stage_n                    73.573310   ±103.21%     58.594629   True
N04D_Stage_a                     4.691568   ±103.97%      6.168789   True
4-6_EIS_2_A                    411.277657   ±737.77%    415.075241   True
4-6_EIS_2_n                      2.270485    ±31.66%      2.180111   True
4-6_EIS_2_a                   5542.742428   ±411.04%   6319.362730   True
constant_d                     366.109755     ±0.18%    366.280923   True
noise_alpha                     20.859894    ±11.96%      1.000000   True

Warnings! (2)
======================================================================
Response tmax for 'Linear_Model_Recharge' > than calibration period.
Response tmax for '4-6_EIS_2' > than calibration period.
```

Out[20]:

```
0.6516507782622496
```



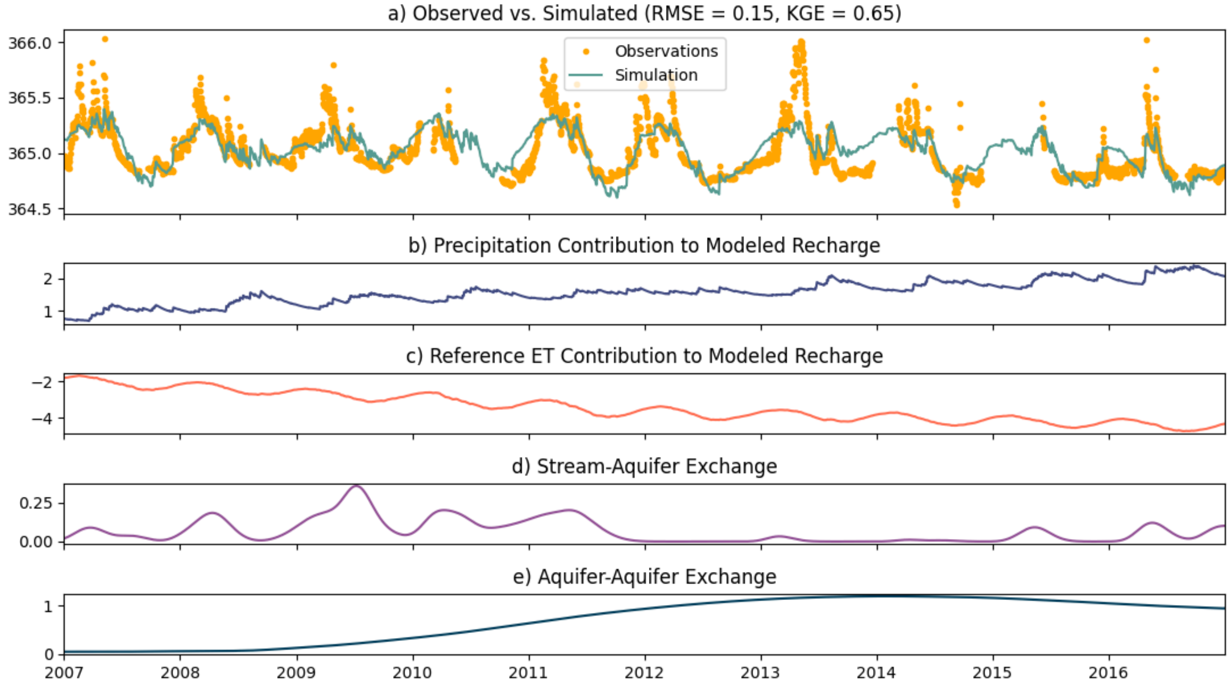
Model Parameters (n_c=13)

name	optimal	stderr
Linear_Model_Recharge_A	1.41	297.57%
Linear_Model_Recharge_alpha	0.92	21.98%
Linear_Model_Recharge_a1	67.73	35.57%
Linear_Model_Recharge_a2	4999.93	430.46%
Linear_Model_Recharge_f	-1.47	11.57%
N04D_Stage_A	1.88	43.36%
N04D_Stage_n	73.57	103.21%
N04D_Stage_a	4.69	103.97%
4-6_EIS_2_A	411.28	737.77%
4-6_EIS_2_n	2.27	31.66%
4-6_EIS_2_a	5542.74	411.04%
constant_d	366.11	0.18%
noise_alpha	20.86	11.96%

In [21]:

```
# Get the contributions of individual stress models
contributions = smdl.get_contributions()
print(contributions)

precip_contr = contributions[0]
ref_et_contr = contributions[1]
stage_et_contr = contributions[2]
EIS_contr = contributions[3]

# Get the observed and simulated time series
observed_smdl = smdl.oseries.series
simulated_smdl = smdl.simulate()

# Create a figure and subplots with different heights
fig, (ax1, ax2, ax3, ax4, ax5) = plt.subplots(5, 1, sharex=True, gridspec_kw={'height_

# Plot the data using my_series as the x-values and corresponding y-values
# Plot the observed and simulated time series on ax1
ax1.plot(observed_smdl.index, observed_smdl, label="Observations", color='#ffa600', li
ax1.plot(simulated_smdl.index, simulated_smdl, label="Simulation", color="#599e94")

# Set the x-axis limits to match the range of my_series
ax1.set_xlim(observed_smdl.index.min(), observed_smdl.index.max())

# Plot data on each subplot
ax2.plot(precip_contr, label='Stream-Aquifer Exchange', color='#444e86')
ax3.plot(ref_et_contr, label='Diffuse Recharge', color='#ff6e54')
ax4.plot(stage_et_contr, label='Reference ET', color='#955196')
ax5.plot(EIS_contr, label='Reference ET', color='#003f5c')

# Add titles to subplots
ax1.set_title("a) Observed vs. Simulated (RMSE = 0.15, KGE = 0.65)")
ax2.set_title("b) Precipitation Contribution to Modeled Recharge")
ax3.set_title("c) Reference ET Contribution to Modeled Recharge")
ax4.set_title("d) Stream-Aquifer Exchange")
ax5.set_title("e) Aquifer-Aquifer Exchange")

ax1.legend(loc='upper center')

# Show the plot
plt.tight_layout()
plt.show()
```

```
[index
2007-01-01    0.762347
2007-01-02    0.759544
2007-01-03    0.756781
2007-01-04    0.754057
2007-01-05    0.751370
                ...
2016-12-27    2.073775
2016-12-28    2.070420
2016-12-29    2.067109
2016-12-30    2.063841
2016-12-31    2.060616
Name: Linear_Model_Recharge (Precip), Length: 3653, dtype: float64, index
2007-01-01   -1.821082
2007-01-02   -1.818328
2007-01-03   -1.818721
2007-01-04   -1.816298
2007-01-05   -1.811659
                ...
2016-12-27   -4.337807
2016-12-28   -4.334399
2016-12-29   -4.331598
2016-12-30   -4.330245
2016-12-31   -4.327034
Name: Linear_Model_Recharge (Evap), Length: 3653, dtype: float64, 2007-01-01    0.019
679
2007-01-02    0.020367
2007-01-03    0.021081
2007-01-04    0.021819
2007-01-05    0.022582
                ...
2016-12-27    0.101118
2016-12-28    0.101213
2016-12-29    0.101292
2016-12-30    0.101355
2016-12-31    0.101404
Freq: D, Name: N04D_Stage, Length: 3653, dtype: float64, 2007-01-01    0.050826
2007-01-02    0.050822
2007-01-03    0.050817
2007-01-04    0.050812
2007-01-05    0.050806
                ...
2016-12-27    0.941546
2016-12-28    0.941299
2016-12-29    0.941051
2016-12-30    0.943710
2016-12-31    0.946369
Freq: D, Name: 4-6_EIS_2, Length: 3653, dtype: float64]
```

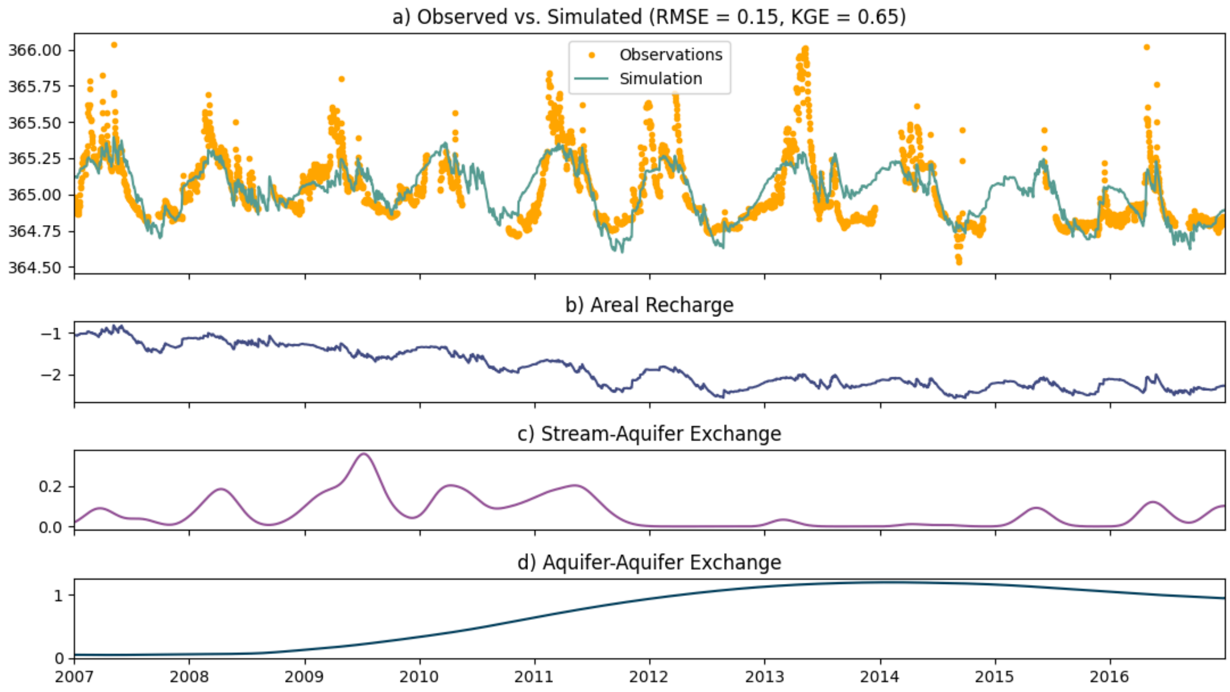


In [22]:

```
recharge_sim = smdl.get_contribution("Linear_Model_Recharge")
#print(recharge_sim)
```

In [23]:

```
# Create a figure and subplots with different heights
fig, (ax1, ax2, ax3, ax4) = plt.subplots(4, 1, sharex=True, gridspec_kw={'height_ratic

# Plot the data using my_series as the x-values and corresponding y-values
# Plot the observed and simulated time series on ax1
ax1.plot(observed_smdl.index, observed_smdl, label="Observations", color='#ffa600', li
ax1.plot(simulated_smdl.index, simulated_smdl, label="Simulation", color="#599e94")

# Set the x-axis limits to match the range of my_series
ax1.set_xlim(observed_smdl.index.min(), observed_smdl.index.max())

# Plot data on each subplot
ax2.plot(recharge_sim, label='Areal Recharge', color='#444e86')
ax3.plot(stage_et_contr, label='Reference ET', color='#955196')
ax4.plot(EIS_contr, label='Reference ET', color='#003f5c')

# Add titles to subplots
ax1.set_title("a) Observed vs. Simulated (RMSE = 0.15, KGE = 0.65)")
ax2.set_title("b) Areal Recharge")
ax3.set_title("c) Stream-Aquifer Exchange")
ax4.set_title("d) Aquifer-Aquifer Exchange")

ax1.legend(loc='upper center')

# Show the plot
plt.tight_layout()
plt.show()
```



Best Model, Model M8 — 3-5 MOR: FlexModel(precip, evap(Gamma)) + precip(DoubleExponential) + EIS Simulated Head(DoubleExponential)

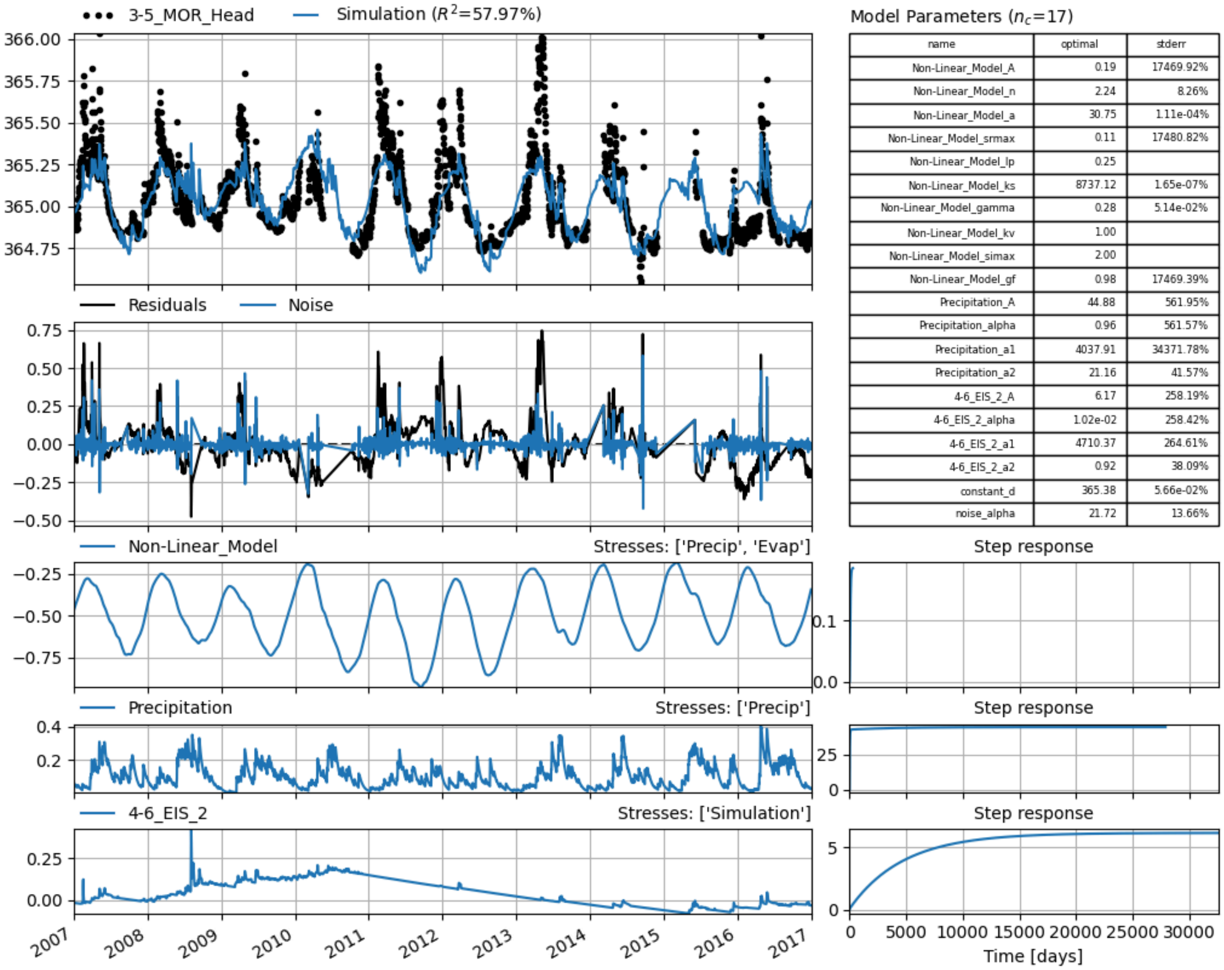
In [24]:

```
ml10c = ps.Model(MOR35_series, name="MOR_10c")

# ----------------------------------------- Stress model settings --------------------
#up=False #Positive values in stress DO NOT increase simulation values

# Adding a recharge model
# ----------------------------------------- Select a recharge model ------------------
rch = ps.rch.FlexModel(gw_uptake=True)
#rch = ps.rch.Berendrecht()
#rch = ps.rch.Linear()
#rch = ps.rch.Peterson()

## ----------------------------------------- Add response function (rfunc) -----------
#rfunc = ps.Exponential() #used for stresses that start to have an immediate effect on
rfunc =ps.Gamma() #most used response function in Pastas, versatile and able to mimic
#rfunc =ps.Hantush() #used for pumping wells
#rfunc =ps.Polder() #intended for the simulation of the response to variations of surf
#rfunc =ps.FourParam() #four parameters that includes the Exponential, Gamma, Hantush
#rfunc =ps.DoubleExponential() #intended for the simulation in a multi-aquifer system,
#rfunc = ps.Edelman() #describing the propagation of an instantaneous water level chan
#rfunc = ps.One() #Instant response with no lag and one parameter d.

#sm1 = ps.StressModel(waterlevel_zero, rfunc, name="EIS_mdl", settings=("flux"))

sm1 = ps.RechargeModel(precip_mm, evap_mm, rfunc = rfunc, name="Non-Linear_Model", rec

ml10c.add_stressmodel(sm1)

ml10c.solve(warmup = warmup, tmin=begin_date, tmax=end_date, noise=False,  report=Fals
ml10c.solve(warmup = warmup, tmin=begin_date, tmax=end_date, noise=True, initial=False

#ml8.plots.results()

## ----------------------------------------- Add response function (rfunc) -----------
#rfunc = ps.Exponential() #used for stresses that start to have an immediate effect on
#rfunc =ps.Gamma() #most used response function in Pastas, versatile and able to mimic
#rfunc =ps.Hantush() #used for pumping wells
#rfunc =ps.Polder() #intended for the simulation of the response to variations of surf
#rfunc =ps.FourParam() #four parameters that includes the Exponential, Gamma, Hantush
rfunc =ps.DoubleExponential() #intended for the simulation in a multi-aquifer system,
#rfunc = ps.Edelman() #describing the propagation of an instantaneous water level chan
#rfunc = ps.One() #Instant response with no lag and one parameter d.

sm2 = ps.StressModel(precip, rfunc, name="Precipitation", settings=("prec"))

ml10c.add_stressmodel(sm2)

ml10c.solve(warmup = warmup, tmin=begin_date, tmax=end_date, report=False, noise=False

ml10c.solve(warmup = warmup, tmin=begin_date, tmax=end_date, initial=False, noise=True

## ----------------------------------------- Add response function (rfunc) -----------
#rfunc = ps.Exponential() #used for stresses that start to have an immediate effect on
#rfunc =ps.Gamma() #most used response function in Pastas, versatile and able to mimic
#rfunc =ps.Hantush() #used for pumping wells
#rfunc =ps.Polder() #intended for the simulation of the response to variations of surf
#rfunc =ps.FourParam() #four parameters that includes the Exponential, Gamma, Hantush
rfunc =ps.DoubleExponential() #intended for the simulation in a multi-aquifer system,
#rfunc = ps.Edelman() #describing the propagation of an instantaneous water level chan
#rfunc = ps.One() #Instant response with no lag and one parameter d.

sm3 = ps.StressModel(EIS_model_norm, rfunc, name="4-6_EIS_2", settings=("prec"))

ml10c.add_stressmodel(sm3)


ml10c.solve(warmup = warmup, tmin=begin_date, tmax=end_date, report=False, noise=False

ml10c.solve(warmup = warmup, tmin=begin_date, tmax=end_date, initial=False, noise=True


ml10c.plots.results(tmax='2017')
#plt.savefig('MOR_rech_fourparam_precip.png')

print(ml10c.stats.kge_2012())

# According to the step response, if it rained continuously at 1 mm/day, it would take
```

```
WARNING: The Time Series '3-5_MOR_Head' has nan-values. Pastas will use the fill_nan
settings to fill up the nan-values.
INFO: Time Series '3-5_MOR_Head': 340 nan-value(s) was/were found and filled with: dr
op.
```

```
Fit report MOR_10c                                   Fit Statistics
===================================================================
nfev    51                     EVP                             6.74
nobs    2909                   R2                              0.07
noise   True                   RMSE                            0.23
tmin    2007-01-01 00:00:00    AIC                        -18311.86
tmax    2016-12-31 00:00:00    BIC                        -18258.08
freq    D                      Obj                             2.67
warmup  1095 days 00:00:00     ___                                 
solver  LeastSquares           Interp.                           No

Parameters (9 optimized)
===================================================================
                             optimal      stderr     initial   vary
Non-Linear_Model_A      1.948017e-02     ±46.33%    0.078012   True
Non-Linear_Model_n      4.890603e-01     ±13.92%    0.789328   True
Non-Linear_Model_a      1.372318e+01    ±119.17%   43.617540   True
Non-Linear_Model_srmax  1.058950e+02     ±45.96%  112.675982   True
Non-Linear_Model_lp     2.500000e-01        ±nan    0.250000  False
Non-Linear_Model_ks     9.534546e+01    ±104.24%  202.375531   True
Non-Linear_Model_gamma  2.355893e+00      ±6.05%    1.907699   True
Non-Linear_Model_kv     1.000000e+00        ±nan    1.000000  False
Non-Linear_Model_simax  2.000000e+00        ±nan    2.000000  False
Non-Linear_Model_gf     9.095038e-09  ±1.26e+09%    0.999895   True
constant_d              3.649923e+02  ±1.16e-02%  365.103878   True
noise_alpha             5.741763e+01     ±19.13%    1.000000   True

Warnings! (1)
===================================================================
Parameter 'Non-Linear_Model_gf' on lower bound: 0.00e+00
Fit report MOR_10c                                  Fit Statistics
==================================================================
nfev    30                     EVP                           50.15
nobs    2909                   R2                             0.50
noise   True                   RMSE                           0.17
tmin    2007-01-01 00:00:00    AIC                       -17762.10
tmax    2016-12-31 00:00:00    BIC                       -17684.42
freq    D                      Obj                            3.21
warmup  1095 days 00:00:00     ___                                
solver  LeastSquares           Interp.                          No

Parameters (13 optimized)
==================================================================
                            optimal      stderr     initial   vary
Non-Linear_Model_A         0.157379  ±47074.14%    0.149453   True
Non-Linear_Model_n         4.100532     ±62.82%    4.110828   True
Non-Linear_Model_a        16.465622     ±69.29%   16.074399   True
Non-Linear_Model_srmax     0.000033  ±1.29e+07%    0.431409   True
Non-Linear_Model_lp        0.250000        ±nan    0.250000  False
Non-Linear_Model_ks     1859.318230  ±2.84e-12%   32.475451   True
Non-Linear_Model_gamma    11.188063  ±6.48e-10%   19.285286   True
Non-Linear_Model_kv        1.000000        ±nan    1.000000  False
Non-Linear_Model_simax     2.000000        ±nan    2.000000  False
Non-Linear_Model_gf        0.995409  ±47073.49%    0.988252   True
Precipitation_A           49.246235     ±69.51%   36.788306   True
Precipitation_alpha        0.011604  ±1.30e+06%    0.092273   True
Precipitation_a1          22.673022   ±1039.67%    9.788488   True
Precipitation_a2          24.991391  ±48676.14%   15.475514   True
constant_d               365.340363  ±2.40e-02%  365.342192   True
noise_alpha               25.053568     ±13.80%    1.000000   True
```

```
Fit report MOR_10c                                    Fit Statistics
====================================================================
nfev    48                     EVP                             58.06
nobs    2909                   R2                               0.58
noise   True                   RMSE                             0.15
tmin    2007-01-01 00:00:00    AIC                         -17824.02
tmax    2016-12-31 00:00:00    BIC                         -17722.43
freq    D                      Obj                              3.14
warmup  1095 days 00:00:00     ___                                  
solver  LeastSquares           Interp.                            No

Parameters (17 optimized)
====================================================================
                                optimal      stderr      initial   vary
Non-Linear_Model_A             0.185465  ±17469.92%     0.164522   True
Non-Linear_Model_n             2.235827      ±8.26%     3.115343   True
Non-Linear_Model_a            30.748170  ±1.11e-04%    21.577687   True
Non-Linear_Model_srmax         0.107391  ±17480.82%     0.000131   True
Non-Linear_Model_lp            0.250000        ±nan     0.250000  False
Non-Linear_Model_ks         8737.121990  ±1.65e-07%   317.793493   True
Non-Linear_Model_gamma         0.282506  ±5.14e-02%    15.311557   True
Non-Linear_Model_kv            1.000000        ±nan     1.000000  False
Non-Linear_Model_simax         2.000000        ±nan     2.000000  False
Non-Linear_Model_gf            0.977746  ±17469.39%     0.987066   True
Precipitation_A               44.878973    ±561.95%    39.573129   True
Precipitation_alpha            0.959365    ±561.57%     0.989955   True
Precipitation_a1            4037.908737  ±34371.78%  3628.136652   True
Precipitation_a2              21.162910     ±41.57%    10.953603   True
4-6_EIS_2_A                    6.166811    ±258.19%     5.726051   True
4-6_EIS_2_alpha                0.010237    ±258.42%     0.010000   True
4-6_EIS_2_a1                4710.369537    ±264.61%  4455.147036   True
4-6_EIS_2_a2                   0.924952     ±38.09%     1.515654   True
constant_d                   365.380260  ±5.66e-02%   365.350004   True
noise_alpha                   21.719880     ±13.66%     1.000000   True

Warnings! (2)
====================================================================
Response tmax for 'Precipitation' > than calibration period.
Response tmax for '4-6_EIS_2' > than calibration period.
0.65522020424356
```



Model Parameters (n_c=17)

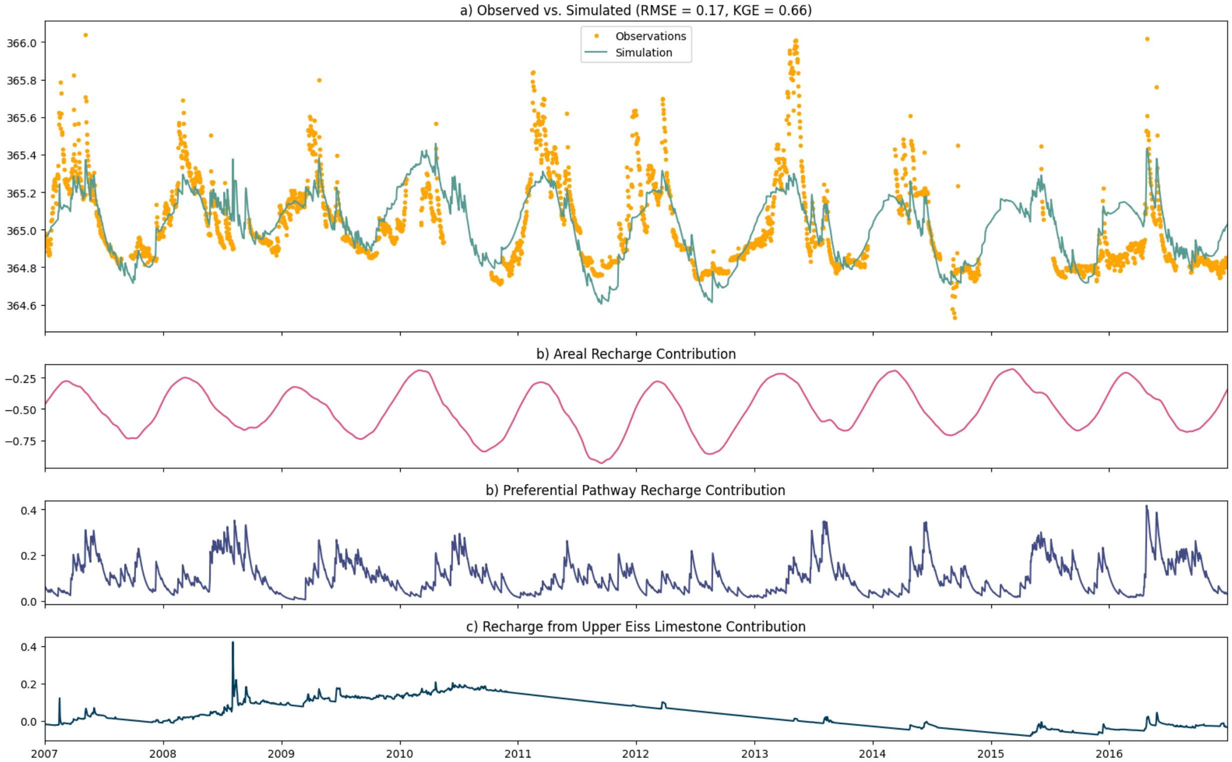
name	optimal	stderr
Non-Linear_Model_A	0.19	17469.92%
Non-Linear_Model_n	2.24	8.26%
Non-Linear_Model_a	30.75	1.11e-04%
Non-Linear_Model_srmax	0.11	17480.82%
Non-Linear_Model_lp	0.25	
Non-Linear_Model_ks	8737.12	1.65e-07%
Non-Linear_Model_gamma	0.28	5.14e-02%
Non-Linear_Model_kv	1.00	
Non-Linear_Model_simax	2.00	
Non-Linear_Model_gf	0.98	17469.39%
Precipitation_A	44.88	561.95%
Precipitation_alpha	0.96	561.57%
Precipitation_a1	4037.91	34371.78%
Precipitation_a2	21.16	41.57%
4-6_EIS_2_A	6.17	258.19%
4-6_EIS_2_alpha	1.02e-02	258.42%
4-6_EIS_2_a1	4710.37	264.61%
4-6_EIS_2_a2	0.92	38.09%
constant_d	365.38	5.66e-02%
noise_alpha	21.72	13.66%

In [25]:

```
# Get the contributions of individual stress models
contributions = ml10c.get_contributions()
print(contributions)

mdl_recharge_contr = contributions[0]
precip_contr = contributions[1]
EIS_contr = contributions[2]

# Get the observed and simulated time series
observed_ml10c = ml10c.oseries.series
simulated_ml10c = ml10c.simulate()

# Create a figure and subplots with different heights
fig, (ax1, ax2, ax3, ax4) = plt.subplots(4, 1, sharex=True, gridspec_kw={'height_ratic

# Plot the data using my_series as the x-values and corresponding y-values
# Plot the observed and simulated time series on ax1
ax1.plot(observed_ml10c.index, observed_ml10c, label="Observations", color='#ffa600',
ax1.plot(simulated_ml10c.index, simulated_ml10c, label="Simulation", color="#599e94")

# Set the x-axis limits to match the range of my_series
ax1.set_xlim(observed_ml10c.index.min(), observed_ml10c.index.max())

# Plot data on each subplot
ax2.plot(mdl_recharge_contr, label='Diffuse Recharge', color='#dd5182')
ax3.plot(precip_contr, label='Diffuse Recharge', color='#444e86')
ax4.plot(EIS_contr, label='Diffuse Recharge', color='#003f5c')
```

```python
# Add titles to subplots
ax1.set_title("a) Observed vs. Simulated (RMSE = 0.17, KGE = 0.66)")
ax2.set_title("b) Areal Recharge Contribution")
ax3.set_title("b) Preferential Pathway Recharge Contribution")
ax4.set_title("c) Recharge from Upper Eiss Limestone Contribution")

ax1.legend(loc='upper center')

# Show the plot
plt.tight_layout()
plt.show()
```

```
[index
2007-01-01   -0.466857
2007-01-02   -0.462742
2007-01-03   -0.458623
2007-01-04   -0.454586
2007-01-05   -0.450596
                ...
2016-12-27   -0.368267
2016-12-28   -0.363398
2016-12-29   -0.358618
2016-12-30   -0.353961
2016-12-31   -0.349462
Name: Non-Linear_Model, Length: 3653, dtype: float64, index
2007-01-01    0.065255
2007-01-02    0.062288
2007-01-03    0.059458
2007-01-04    0.056759
2007-01-05    0.054184
                ...
2016-12-27    0.035241
2016-12-28    0.033750
2016-12-29    0.032328
2016-12-30    0.030972
2016-12-31    0.029678
Name: Precipitation, Length: 3653, dtype: float64, 2007-01-01   -0.018277
2007-01-02   -0.018442
2007-01-03   -0.018607
2007-01-04   -0.018771
2007-01-05   -0.018936
                ...
2016-12-27   -0.032875
2016-12-28   -0.033421
2016-12-29   -0.033713
2016-12-30   -0.033857
2016-12-31   -0.033973
Freq: D, Name: 4-6_EIS_2, Length: 3653, dtype: float64]
```


a) Observed vs. Simulated (RMSE = 0.17, KGE = 0.66)
Observations
Simulation
366.0
365.8
365.6
365.4
365.2
365.0
364.8
364.6
b) Areal Recharge Contribution
−0.25
−0.50
−0.75
b) Preferential Pathway Recharge Contribution
0.4
0.2
0.0
c) Recharge from Upper Eiss Limestone Contribution
0.4
0.2
0.0
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016