

TEXTE

91/2026

Abschlussbericht

PFLAST

**Aufbau einer Daten- und Modellierungsplattform zur
Berechnung von Stickstoffflüssen in der Landwirtschaft
sowie zur Bewertung von Minderungsmaßnahmen**

von:

Müller, Matthias, Heimbürger, Jan
PIKobytes GmbH, Dresden

Herausgeber:

Umweltbundesamt

TEXTE 91/2026

REFOPLAN des Bundesministeriums Umwelt,
Naturschutz, nukleare Sicherheit und Verbraucherschutz

Forschungskennzahl 3721 51 203 0

Abschlussbericht

PFLAST

Aufbau einer Daten- und Modellierungsplattform zur
Berechnung von Stickstoffflüssen in der Landwirtschaft
sowie zur Bewertung von Minderungsmaßnahmen

von

Müller, Matthias, Heimbürger, Jan
PIKobytes GmbH, Dresden

Im Auftrag des Umweltbundesamtes

Impressum

Herausgeber

Umweltbundesamt
Wörlitzer Platz 1
06844 Dessau-Roßlau
Tel: +49 340-2103-0
Fax: +49 340-2103-2285
buergerservice@uba.de
Internet: www.umweltbundesamt.de

Durchführung der Studie:

PIKobytes GmbH
Zittauer Straße 16
01099 Dresden

Abschlussdatum:

Dezember 2025

Redaktion:

Fachgebiet II 4.1 – Grundsatzfragen der Luftreinhaltung
Andreas Eisold

DOI:

<https://doi.org/10.60810/openumwelt-8252>

ISSN 1862-4804

Dessau-Roßlau, Juni 2026

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autorinnen*Autoren.

Kurzbeschreibung: PFLAST

Gegenstand des Vorhabens „PFLAST“ war die Überführung eines Modells zur Stickstoffflächenbilanzierung auf landwirtschaftlichen Flächen auf Kreisebene nach der Methode von Häußermann et al. (2019)¹ an das Umweltbundesamt (UBA). Das Bilanzierungsmodell wurde neu implementiert und in eine neue Systemarchitektur überführt, die künftig am UBA betrieben, gewartet und fortgeschrieben werden kann.

Die wesentlichen Eingangsdaten stammen aus öffentlich zugänglichen Datenquellen. Diese Daten werden zunächst jahresscharf aufbereitet und versionierbar in einer definierten Datenstruktur abgelegt. Auf dieser Datengrundlage findet dann die eigentliche Bilanzierungsrechnung statt, die die Stickstoff-Flächenbilanzüberschüsse auf landwirtschaftlichen Flächen auf Ebene der Kreise liefert.

Im Zuge der Bearbeitung wurde der Rechenkern des Modells in die Programmiersprache Python überführt und sämtliche Rechenwege im Detail validiert. Dieser Rechenkern ist auf die derzeitigen Bedürfnisse in der Bilanzierungsrechnung zugeschnitten und strukturell auf künftige Änderungsbedarfe in den Rechenvorschriften und Eingangsdaten vorbereitet.

Die notwendigen Eingangsdaten und deren Vorverarbeitungsschritte wurden aufgrund ihres Umfangs zunächst nur in Teilen analysiert und für einzelne Eingangsgrößen ebenfalls in Python neu implementiert. Künftige Entwicklungsaktivitäten sollten die hier begonnenen Arbeiten fortführen und abschließen.

Abstract: PFLAST

The aim of the 'PFLAST' project was to transfer a model for nitrogen balancing at district level based on the method developed by Häußermann et al. (2019)¹ to the German Environment Agency (UBA). This balancing model was reimplemented and transferred to a new sustainable system architecture that can be operated and maintained by UBA staff.

The main input data comes from publicly available data sources. This data is pre-processed and harmonized to an annual basis and stored in a defined data structure in a versionable format. Based on that data, the actual balancing calculation, which provides the nitrogen area balance surpluses on agricultural land, is then carried out.

In the course of the project, the model's 'core' was transferred to the Python programming language, and all calculation methods were validated in detail. This new implemented core is tailored to the current requirements of the nitrogen balancing calculation and structurally prepared for future changes in the calculation rules and input data.

Due to their quantity, the necessary input data and their pre-processing steps were only analysed in parts. The project has only created an inception to reimplement these pre-processing steps in Python which needs to be continued in follow-up activities.

¹ <https://www.umweltbundesamt.de/publikationen/stickstoff-flaechenbilanzen-fuer-deutschland>

Inhaltsverzeichnis

Abbildungsverzeichnis.....	9
Tabellenverzeichnis.....	9
Abkürzungsverzeichnis.....	10
Zusammenfassung.....	11
Summary.....	14
Projektkontext und Zielstellung.....	17
1 Darstellung des IST-Zustands der Implementierung.....	19
1.1 Aufbau und Implementierung.....	19
1.2 Räumliche und zeitliche Granularität.....	19
1.3 Statistisches Mengengerüst und Eingangsdaten.....	20
1.4 Modellpflege und Weiterentwicklung.....	20
2 Definition des SOLL-Zustandes.....	21
2.1 Szenarien, Modelllauf und Basislauf.....	21
2.2 Komponenten des Bilanzierungsmodells.....	21
2.3 Modellpflege, Versionen und Versionierung des Bilanzierungsmodells.....	23
2.4 Konzept zur Modellversionierung.....	24
2.5 Mengengerüst.....	25
2.6 Systemkonzeption.....	26
2.6.1 Datenhaltung.....	26
2.6.1.1 Anforderungen und Randbedingungen.....	26
2.6.1.2 Vorzugsvariante.....	27
2.6.1.3 Alternativen.....	27
2.6.2 Programmierung des Rechenkerns.....	27
2.6.2.1 Anforderungen und Randbedingungen.....	27
2.6.2.2 Vorzugsvariante.....	27
2.6.2.3 Alternativen.....	28
2.6.3 Versionierung.....	28
2.6.3.1 Anforderungen und Randbedingungen.....	28
2.6.3.2 Vorzugsvariante.....	28
2.6.3.3 Varianten zur Umsetzung.....	28
3 Durchführung und Umsetzung.....	30
3.1 Allgemeines Vorgehen.....	30
3.2 Bereitstellung einer Referenzimplementierung des Bilanzierungsmodells.....	30

3.2.1	N Referenzimplementierung 07/2023	30
3.2.2	Referenzimplementierung Oktober/2024	33
3.3	Entwicklung von Werkzeugen zur Analyse der Referenzimplementierung.....	33
3.3.1	Extraktionswerkzeuge.....	33
3.3.2	Werkzeuge zur Konsistenzprüfung	36
3.3.2.1	Werkzeug „compare_pflast_files“	37
3.3.2.2	Werkzeug „formula_analysis“	39
3.3.3	Werkzeuge zur Nachführung und Fehlerbehebung in der Referenzimplementierung	42
3.3.3.1	Werkzeug create_patches	43
3.3.3.2	Werkzeug patch_excel_file.....	43
3.3.4	Werkzeuge zur Analyse und Veranschaulichung des Rechenweges	43
3.4	Umsetzung des Rechenkerns zur Stickstoffbilanzierung	43
3.4.1	Überführung der Rechenvorschriften aus der Referenzimplementierung in Python-Code	43
3.4.2	Technologiestack	44
3.4.3	Struktur des Softwareprojekts.....	44
3.4.4	Workspace	45
3.4.4.1	Workspace-Struktur.....	45
3.4.4.2	Eingangsdaten.....	47
3.4.5	Implementierung der Berechnung.....	48
3.5	Entwicklung von Werkzeugen zur Validierung des Rechenkerns	48
3.5.1	Werkzeug result_comparison	48
3.6	Entwicklung von Werkzeugen zur Analyse der Datenvorverarbeitung	51
3.6.1	Analyse von Vorberechnungsdateien	51
3.6.2	Entwicklung von Werkzeugen zur Extraktion von Daten.....	52
3.7	Entwicklung von Rechenmodulen zur Datenvorverarbeitung.....	52
3.7.1	Abfrage von Rohdaten	53
4	Projektergebnisse.....	54
4.1	Softwareprojekt zur Stickstoffflächenbilanzierung.....	54
4.2	Validierung und technische Dokumentation der Referenzimplementierung des Modells zur Stickstoffbilanzierung.....	54
4.3	Fließschema	54
4.4	Funktionen und Werkzeuge zur Datenvorverarbeitung	55
5	Weiterentwicklung und funktionale Erweiterungen.....	56
5.1	Notwendige Schritte zur Vervollständigung der Neuimplementierung	56

5.1.1	Datenvorverarbeitung	56
5.1.2	Abschließende Prüfung der primären Datenquellen.....	56
5.2	Verstetigung der Modellpflege am UBA	56
5.3	Perspektiven zur funktionalen Weiterentwicklung	57
5.3.1	Sensitivitätsanalysen.....	57
5.3.2	Geografische Auswertungen und Karten.....	57
6	Quellenverzeichnis	58

Abbildungsverzeichnis

Abbildung 1:	Struktureller Aufbau eines Workspaces für das Bilanzierungsmodell	22
Abbildung 2:	Ablaufschema zur Erstellung neuer Hauptversionen und abgeleiteter Varianten.....	24
Abbildung 3:	Struktur der Hauptrechenmappe	31
Abbildung 4:	Tabellenblatt 2019 ber	32
Abbildung 5:	Schematische Darstellung zur mehrstufigen Ableitung von Zwischengrößen zur N-Bilanzierung.	34
Abbildung 6:	Überblick der Bereiche in den Berechnungsblättern	36
Abbildung 7:	Benutzeroberfläche <i>compare_pflast_files</i>	39
Abbildung 8:	Beispiel für eine Formelvergleichsübersicht	41
Abbildung 9:	Beispiel eine Formeltabelle für eine Spalte.....	42
Abbildung 10:	Verzeichnisstruktur des Softwareprojektes	44
Abbildung 11:	Workspace-Struktur	46
Abbildung 12 :	Benutzeroberfläche <i>result_comparison</i>	49
Abbildung 13:	Ausschnitt aus <i>IN_diff_percent.html</i>	50
Abbildung 14:	Fließschema (Ausschnitt).....	55

Tabellenverzeichnis

Tabelle 1:	Mengengerüst zur N-Flächenbilanzierung (Abschätzung)	26
Tabelle 2:	Relevante Tabellenblätter in der Referenzimplementierung...32	
Tabelle 3:	Einstellmöglichkeiten des Vergleichswerkzeuges	38
Tabelle 4:	Übersicht über Dateiausgabe des Ergebnisvergleichs.....	50

Abkürzungsverzeichnis

Abkürzung	Erläuterung
API	Application Programming Interface, Programmierschnittstelle, auch Schnittstelle für maschinenlesbare Datenabruf
CSV	Comma-separated values, ein textbasiertes Dateiformat zur Speicherung tabellarisch strukturierter Daten
GIS	Geografisches Informationssystem
GraphML	XML-basiertes Dateiformat für die Repräsentation von Graphen
HTML	Hypertext Markup Language, Auszeichnungssprache für Webseiten
IDE	Integrated Development Environment, integrierte Entwicklungsumgebung
JLU	Justus-Liebig-Universität Gießen
JSON	JavaScript Object Notation, ein textbasiertes Dateiformat zur Speicherung hierarchisch strukturierter Daten
PDF	Portable Document Format, ein Dateiformat für Dokumente
PFLAST	Plattform landwirtschaftliche Stoffflüsse
UBA	Umweltbundesamt

Zusammenfassung

Das Projekt PFLAST verfolgte das Ziel, die bestehende, über viele Jahre gewachsene Excel-basierte Berechnung der Stickstoff-Flächenbilanzierung (N-Bilanz) nach Häußermann et al. (2019) in eine moderne, leistungsfähige und erweiterbare Neuentwicklung zu überführen. Eine bisher genutzte Implementierung, die vollständig in Microsoft Excel realisiert war, erwies sich mit wachsender Komplexität des Bilanzierungsmodells als kaum noch wartbar und fehleranfällig. Aufgrund extrem langer Rechenzeiten sowie erforderlicher manueller Interaktionen im Berechnungsprozess, war ein produktives Arbeiten mit den Excel-Rechenmappen kaum noch möglich.

Sowohl Eingangsdaten, Vorverarbeitungsschritte als auch die eigentliche Bilanzierung waren auf zahlreiche sehr komplexe und große Rechenmappen verteilt, deren interne Verknüpfungen und Berechnungsschritte oft nur über manuelle Analysen nachvollziehbar waren. Es existierte nur eine minimale Dokumentation und das Wissen über die korrekte Bedienung und Weiterentwicklung lag bei einer einzelnen Person.

Vor diesem Hintergrund wurde im Projekt in mehreren Arbeitsschwerpunkten das Bilanzierungsmodell neu implementiert:

1. Die Feststellung des IST-Zustandes der bisherigen Implementierung
2. Die Konzeption einer Systemarchitektur, die typischen Tätigkeiten im Modellbetrieb und seiner Weiterentwicklung adäquat Rechnung trägt
3. Die Neuprogrammierung des Rechenkerns in der Programmiersprache Python und begleitendes Reverse-Engineering der vorhandenen Excel-basierten Rechenmappen
4. Die Analyse und teilweise Neuimplementierung von Vorverarbeitungsschritten
5. Die im Zuge der IST-Analyse und der Neuprogrammierung als erforderlich gewordene Validierung, Korrektur und Nachführung der Excel-basierten Referenzimplementierung

Ein Teil dieser Arbeiten – insbesondere zum Reverse-Engineering, der Analyse umfangreicherer Vorverarbeitungsschritte sowie der Validierung, Korrektur und Nachführung der Referenzimplementierung – wurden erst im Projektverlauf offenbar und zwingend erforderlich, um das übergeordnete Projektziel zu erreichen. Zudem wurde parallel zur Neuimplementierung an der alten Version weitergearbeitet und in diesem Zuge zusätzliche Nachführungen während der Projektbearbeitung erforderlich.

Herausforderungen im Projekt waren insbesondere die große Komplexität der zu analysierenden Dateien, die unvollständige Trennung von Vorverarbeitung und Bilanzierung, die fehlende Dokumentation der Rechenmappen. Die einzige Möglichkeit zur Neuimplementierung war das schrittweise, manuelle Reverse-Engineering der vorhandenen Rechenmappen und die Klärung von Befunden und Rückfragen mit dem bisherigen Hauptentwickler.

Um den bestehenden Rechenweg der Excel-Implementierung reproduzierbar zu verstehen, wurden zuerst umfangreiche Werkzeuge entwickelt, die es ermöglichen, Konstanten, Parameter, numerische Eingangsdaten und strukturelle Verknüpfungen automatisiert aus den Rechenmappen zu extrahieren.

Besonders wichtig waren darüber hinaus Werkzeuge, die die zahlreichen teils heterogenen Formeln der Excel-Berechnungsblätter miteinander vergleichen, um Formelvarianten und abweichende Rechenvorschriften zu detektieren. Während Formeln weiterhin manuell interpretiert und nachimplementiert werden mussten, konnten Datenflüsse und Abhängigkeiten

mit diesen Werkzeugen automatisiert analysiert werden. In diesem Prozess wurden strukturelle Unstimmigkeiten und Fehler in den Rechenwegen identifiziert, die dann in den Referenzrechenmappen konsistent und nachvollziehbar korrigiert werden mussten. Erst durch diese systematische Aufarbeitung gelang es, eine konsolidierte und korrigierte Referenz zu erstellen, gegen die der neu programmierte Python-Rechenkern abschließend validiert werden konnte. Die gefundenen Fehler wurden zudem quantitativ analysiert, um festzustellen, ob die bisherigen Bilanzierungsrechnungen dadurch maßgeblich verfälscht wurden. Da die festgestellten Gesamtfehler letztendlich nur sehr gering waren, wurden die bisherigen Daten auch vom nachkorrigierten Modell erneut bestätigt.

Die Neuimplementierung des Rechenkerns erfolgte als modulares Python-Projekt, das eine klare Trennung zwischen Konfiguration, Eingangsdaten, Konstanten, Zwischenergebnissen und finalen Bilanzierungsdaten vorsieht. Es wurde ein Workspace-Konzept eingeführt, in dem jeder Berechnungslauf in einer definierten Verzeichnisstruktur stattfindet. Diese umfasst standardisierte Verzeichnisse für Eingangsdaten, Logs, Konfigurationen und Bilanzierungsergebnisse.

Alle für die Bilanzierung nötigen Parameter, Koeffizienten und tabellarischen *Konstanten* werden im Konfigurationsbereich in Form von JSON- und CSV-Dateien verwaltet. Diese textbasierten Dateiformate lassen sich nicht nur mit Standardwerkzeugen leicht bearbeiten und lesen. Sie erlauben es auch, gängige Versionskontrollsysteme für die Fortschreibung oder – sofern erforderlich – zur nachvollziehbaren Korrektur des Datenbestandes einzusetzen. Auch die jährlich variierenden *Eingangsdaten* liegen in einheitlich benannten und strukturierten CSV-Dateien für jedes Kalenderjahr vor und beinhalten umfassende Angaben zu landwirtschaftlichen Nutzungen, Erträgen und Tierbeständen. Aus diesen Gesamtdaten werden vom Rechenkern schließlich die Flächenbilanzen berechnet und zusammen mit Zwischenergebnissen und Log-Dateien in den dafür vorgesehenen Verzeichnissen abgelegt.

Ein wesentliches Ergebnis des Projektes ist ein funktionsfähiger, vollständig in Python implementierter Rechenkern für die Stickstoff-Flächenbilanzierung. Dieser erlaubt reproduzierbare, versionierte Modellläufe, eine klare Abgrenzung zwischen Basislauf und Szenarien sowie eine strukturierte Weiterentwicklung des Modells.

Nach der Fertigstellung und Prüfung des Rechenkerns war die Analyse und Nachimplementierung der Datenvorverarbeitung ein weiterer Aufgabenschwerpunkt. Im Zuge der Vorverarbeitung werden die jährlich variierenden Eingangsdaten aus der Officialstatistik schrittweise harmonisiert und transformiert, bis sie schließlich in der Form vorliegen, wie sie für die Bilanzierung erforderlich ist. Diese Vorberechnungen waren auf eine Vielzahl unterschiedlicher und ebenfalls undokumentierter Rechenmappen verteilt. Wiederkehrende Rechenschritte darin sind Lückenschließungen in Zeitreihen, die Harmonisierung heterogener Gebietszuschnitte sowie die Umrechnung einzelner Datenquellen auf die im Bilanzierungsmodell definierten Kreisregionen.

Für die Analyse und Neuimplementierung wurden mehrere spezialisierte Python-Werkzeuge entwickelt. Weiterhin wurde untersucht, ob sich die Quelldaten künftig automatisiert über maschinenlesbare Schnittstellen bzw. Webdienste beziehen lassen, um die Fortschreibung der Daten künftig stärker zu automatisieren. Gleichzeitig zeigte sich, dass nicht alle Eingangsdaten automatisiert beschafft werden können, da einige Daten weiterhin manuelle Übertragungen

oder Umformatierungen erfordern. Zwar konnte für einige Datenquellen (destatis.de und regionalstatistik.de) ein automatisierter Datenabruf in Aussicht gestellt werden. Im Einzelfall verhindern jedoch medienbruchhafte und historisch gewachsene Datenlieferwege für einzelne Datensätze die vollständige Automatisierung der Datenabrufs-Pipeline.

Der Prozess der Analyse, Konsolidierung und Neuimplementierung der Vorberechnungen konnte lediglich begonnen und teilweise umgesetzt werden. Eine vollständige Analyse und Nachbildung der Vorberechnungen war im Zuge der Projektdurchführung nicht mehr möglich. Diese Arbeiten müssen im Rahmen künftiger Entwicklungsaktivitäten fortgeführt und abgeschlossen werden, um das neu implementierte Modell produktiv und effizient nutzen zu können.

Die erzielten Projektergebnisse sind dennoch sehr umfangreich: Die Migration des Rechenkerns ist ein grundlegender Schritt, um die Stickstoff-Flächenbilanzierung nach Häußermann et al. (2019) auch künftig nutzen und weiterentwickeln zu können. Durch die Umstellung können große Teile des Modellcodes mit automatisierten Tests abgedeckt werden, die die Fehleranfälligkeit stark reduzieren. Durch den Einsatz einer modernen Programmiersprache und leistungsfähiger Bibliotheken zur Datenverarbeitung bieten sich zudem neue Möglichkeiten zu einer umfassenderen Validierung der Bilanzierungsergebnisse in Form von regelmäßig durchführbaren Sensitivitätsanalysen. In Anbetracht der Projektaufwände für Analyse und Reverse-Engineering wäre es sehr sinnvoll gewesen, dieses Projekt schon deutlich früher durchzuführen, als die Excel-Rechenmappen noch weniger komplex und kompakter waren.

Mit der Migration ist auch die Komplexität des Modells wieder besser beherrschbar geworden. Der neue Rechenkern ist ein belastbares Fundament für die Weiterentwicklung des bestehenden N-Bilanzierungsmodells und auch die perspektivische Erweiterung auf weitere Stoffflüsse wie Phosphor oder Kohlenstoff. Gleichzeitig wurden wesentliche Anforderungen für einen zukünftig stärker automatisierten und fachlich erweiterbaren Modellbetrieb umgesetzt, auch wenn einzelne Aufgaben – insbesondere im Bereich der Datenvorverarbeitung und der Stabilisierung der operativen Systemumgebung – über das Projekt hinaus weiterzuführen sind. Die neue Systemarchitektur ermöglicht künftig auch die modulare Erweiterung der Modellfunktionalitäten, wie beispielsweise die Berechnung von Minderungspotenzialen von Maßnahmenoptionen auf nationaler Ebene, auf Basis des vorhandenen Datengerüsts und der umgesetzten Vorverarbeitungsschritte.

Summary

The PFLAST project aimed to transform the existing Excel-based calculation of nitrogen area balancing (N balance) by Häußermann et al. (2019) into a modern, powerful, and extensible software. A previously used implementation, which was realized in Microsoft Excel and had grown over many years, proved difficult to maintain and prone to errors as the complexity of the balance model increased. Due to extremely long calculation times and the necessary manual interactions in the calculation process, it was hardly possible to work productively with the Excel spreadsheets.

Input data, preprocessing steps, and the actual accounting were distributed across numerous highly complex and large Excel sheets, whose internal links and calculation steps could often only be traced through manual analysis. There was minimal documentation and knowledge of the correct operation and further development was held by a single person.

Against this background, the balance model was reimplemented in the project with several key areas of focus:

1. Initial assessment of the Excel-based implementation
2. Design of a system architecture that adequately considers typical activities in model operation and future development
3. Reprogramming of core calculations in the Python programming language and reverse-engineering of existing Excel-based calculation
4. Analysis and partial reimplementation of required preprocessing steps
5. Validation of the Excel-based reference implementation and application of corrections and fixes based on identified errors and issues.

Some of this work – in particular, reverse engineering, the analysis of more extensive preprocessing steps, and the validation, correction, and updating of the reference implementation – only became apparent during the project and was absolutely necessary to achieve the overall project goal. As work on the old version continued in parallel with the new implementation, additional updates during the course of the project were required.

Challenges in the project included the high complexity of the files to be analysed, the tight entanglement of preprocessing and core calculations, and the lack of documentation for the Excel spreadsheets. The only option for reimplementation was to gradually reverse engineer the existing calculation sheets and clarify findings and queries with the previous lead developer.

To migrate the existing Excel implementation in a reproducible manner, comprehensive tools were first developed that enable constants, parameters, numerical input data, and structural links to be automatically extracted from the Excel spreadsheets.

In addition, tools that compare the numerous, sometimes heterogeneous formulas in the Excel calculation sheets with each other to detect formula variants and deviating calculation rules proved particularly important. While formulas still had to be interpreted and reimplemented manually, data flows and dependencies could be analysed automatically with these tools. This process identified some structural inconsistencies and errors in the calculation methods, which then had to be corrected consistently and comprehensibly in the reference workbooks. Only through this systematic review it was possible to create a consolidated and corrected reference against which the newly programmed Python calculation engine could be finally validated. The

errors found were also analysed quantitatively to determine whether they had significantly distorted the balance calculations in the past. Since the total errors identified were ultimately very minor, the previous data was reconfirmed by the corrected model.

The calculation core was reimplemented as a modular Python project that provides a clear separation between configuration, input data, constants, intermediate results, and final accounting data. A workspace concept was introduced in which each calculation run takes place in a defined directory structure. This includes standardised directories for input data, logs, configurations, and accounting results.

All parameters, coefficients, and tabular constants required for balancing are managed in configuration files encoded in JSON and CSV. These text-based file formats are easy to edit and read with standard tools and work nicely with common version control systems to be used for updating or, if necessary, for traceable correction of the data stock. The annually varying input data is also available in uniformly named and structured CSV files for each calendar year and contains comprehensive information on agricultural use, yields, and livestock. The calculation engine uses this aggregate data to calculate the area balances and stores them together with interim results and log files in the designated directories.

A key result of the project is a functional calculation engine for nitrogen area balancing, implemented entirely in Python. This allows for reproducible, versioned model runs, a clear distinction between base runs and scenarios, and facilitates the further development of the balance model.

With a completed and tested calculation engine in place, the analysis and subsequent implementation of data preprocessing was another key task. The existing calculations were distributed across a large number of different and also undocumented calculation sheets. During the preprocessing steps, the annually varying input data from official statistics are gradually harmonized and transformed until they are finally available in the form required for balancing. Recurring calculation tasks include closing gaps in time series, harmonising heterogeneous area divisions, and converting individual data sources to the districts and regions defined in the balance model.

Several specialized Python tools were developed for the analysis and reimplementation. Furthermore, it was investigated whether the source data could be obtained automatically in the future via machine-readable interfaces or web services to automate the updating of data to a greater extent. At the same time, it became apparent that not all input data can be obtained automatically, as some data still requires manual transfer or reformatting. Although automated data retrieval was promised for some data sources (destatis.de and regionalstatistik.de in particular), media discontinuity and historically grown data delivery channels for individual data sets prevent the complete automation of the data retrieval pipeline.

A complete analysis and reimplementation of the preprocessing calculations was not possible within this project. Nevertheless, this work was started and produced an initial set of analysis results and a small set of preprocessing functions and tools in Python. This effort must be continued and completed as part of future development activities to be able to use the newly implemented model productively and efficiently.

Overall, the achieved project results are quite extensive: the migration of the calculation core of the balance model is a fundamental step toward being able to continue using and further developing nitrogen area balancing from Häußermann et al. (2019) in the future. The technological switch allows large parts of the model code to be covered by automated tests, which greatly reduce the susceptibility to trivial and non-trivial errors. The use of a modern programming language and powerful libraries for data processing also offers new opportunities for more comprehensive validation of the balancing results in the form of regularly performed sensitivity analyses. Considering the re-engineering and validation efforts, it would have made a lot of sense to carry out this project at a much earlier stage, when the Excel implementation was less complex and distributed over fewer spreadsheets and files.

The migration has reduced the complexity and restored the manageability of the balance model. The new computing core provides a robust foundation for the further development of the existing N-balancing model and for its anticipated expansion to include other material flows such as phosphorus or carbon. At the same time, key requirements for a more automated and technically expandable model operation in the future have been implemented, even though individual tasks – particularly around data preprocessing and stabilisation of the operational system environment – will have to be continued after this project has ended. The new system architecture supports such activities. It enables the modular expansion of model functionalities in the future, such as the calculation of reduction potentials of policy options at national level, based on the existing data framework and the implemented and analysed pre-processing steps.

Projektkontext und Zielstellung

Bestimmte Stoffeinträge aus der industriellen Landwirtschaft belasten Mensch und Umwelt. Reaktive Stickstoffverbindungen stehen aufgrund hoher regionaler und nationaler Belastungen besonders im Fokus. Stoffe wie Phosphor und Kohlenstoff, in Form der Treibhausgase Methan und Kohlenstoffdioxid, spielen ebenfalls eine bedeutende Rolle. Um negative Umweltwirkungen bewerten und reduzieren zu können, müssen die Stoffflüsse über das gesamte landwirtschaftliche Produktionssystem und die nachgelagerten Belastungspfade hinweg ganzheitlich erfasst werden.

Im Zuständigkeitsbereich des Umweltbundesamtes (UBA) ist ein wesentlicher Baustein der Stoffflussberechnungen für Stickstoff die regionale Stickstoffflächenbilanzierung auf landwirtschaftlichen Flächen. Diese ist ein zentraler Baustein für verschiedene UBA-Facheinheiten zur Erfüllung von Berichtspflichten und Aufgaben der Politikberatung sowie zur Information der Öffentlichkeit. Im Auftrag und mit Unterstützung des UBA entwickelte eine Arbeitsgruppe der Justus-Liebig-Universität Gießen (JLU) um Herrn Dr. Bach in den vergangenen Jahren ein detailliertes Modell, das aus zahlreichen externen Eingangsdaten die Stickstoffflächenbilanzen auf landwirtschaftlichen Flächen für Deutschland auf Kreisebene berechnet. Darauf aufbauend entstanden und entstehen weitere Modellbausteine, die weitere Stoffgruppen und entsprechende Minderungspotenziale verschiedener Maßnahmen und Maßnahmenpakete auf nationaler Ebene betrachten. Ziel dieses Vorhabens war es, die bestehenden Modellbausteine in einem konsistenten System modular zu verknüpfen.

Das an der JLU in Microsoft Excel betriebene Modell wurde im Rahmen des Vorhabens an das UBA überführt. In diesem Zuge wurde das Modell vollständig neu implementiert, da die Excel-basierte Implementierung fehleranfällig und schwer wartbar ist und von externen Dritten nicht bedient und erweitert werden kann. Die Neuimplementierung sollte in einer zeitgemäßen, weit verbreiteten Programmiersprache erfolgen und die zusätzlich benötigte Datenhaltung zu Eingangsdaten und Bilanzierungsergebnissen sollte neu konzipiert und umgesetzt werden.

Ursprünglich sollten die Modellstrukturen der JLU so in eine zukünftig am UBA gehostete Systemarchitektur überführt werden, dass

1. die Ergebnisse der Stickstoffflächenbilanzierung aus dem Jahr 2019 für die Zeitreihe 1995 bis 2017 auf Kreisebene mit dem neuen System reproduziert werden können,
2. die notwendigen Eingangsdaten, wenn möglich, direkt über Schnittstellen zu den externen Datenquellen eingebunden werden, so dass Eingangsdaten auf den neusten Stand aktualisiert werden können und damit die Zeitreihe bis zu einem aktuellen Jahr fortgeschrieben werden kann,
3. die Berechnung von Maßnahmenminderungspotenzialen und Projektionen reproduziert werden kann und
4. die Ergebnisse in einer solchen Form ausgegeben werden können, dass sie von bereits existierenden Datenbanken und Modellen am UBA direkt oder zur Weiterverarbeitung verwendet werden können und sich mit geeigneter Software visualisieren lassen.

Nach Projektbeginn konnte die bestehende Implementierung erstmalig in Augenschein genommen und analysiert werden. Im Zuge der initialen Bestandsaufnahme und der späteren Arbeit an der Neuimplementierung wurden weitere Rahmenbedingungen bekannt, die eine wiederholte Anpassung der ursprünglichen Projektziele und des Arbeitsplans erforderten. Hier sind insbesondere zu nennen:

1. Eine fehlende detaillierte Dokumentation der bisherigen Rechenmappen, die mit Hilfe der AG Bach projektbegleitend mit aufgebaut und kleinteilig gegen die Referenzimplementierung in Excel geprüft werden musste.
2. Das Fehlen eines konsolidierten Arbeitsstandes der Referenzimplementierung, der als Grundlage für die Analyse und die Neuimplementierung dienen konnte. Dieser musste in den ersten Projektmonaten durch die AG Bach erst aus verschiedenen bestehenden Versionen erzeugt werden.
3. Die wiederholte, projektbegleitende Fortschreibung der Referenzimplementierung, sodass Rechenwege und Zwischenstände in der Neuimplementierung mehrfach geprüft und angepasst werden mussten.
4. Die Identifikation struktureller Inkonsistenzen in den Rechenmappen, die eine aufwändige Prüfung und Nachkorrektur der Referenzimplementierung in Excel erforderlich machten.
5. Eine besonders hohe Komplexität der Rechenmappen, die zu häufigen Abstürzen von Excel und / oder zu unvollständigen Berechnungsergebnissen führte.
6. Die Übergabe vieler zusätzlicher bis dato undokumentierter Rechenmappen, in die Vorverarbeitungsschritte in der Datenaufbereitung ausgelagert waren und die zusätzlich analysiert werden mussten.
7. Die nur teilweise Verfügbarkeit von Eingangsdaten über maschinenlesbare Schnittstellen, die einen vollautomatisierten Datenbezug von externen Anbietern*Anbieterinnen verhindern.
8. Die in der Vergangenheit häufigere Umstellung von Lieferwegen und das Fehlen / Wegfallen bestimmter Eingangsdaten, die einer stärkeren Automatisierung in der Datenaktualisierung ebenfalls im Weg stehen.
9. Die projektbegleitende Anbahnung einer engeren Kooperation mit dem Thünen-Institut zur gemeinsamen Nutzung von Eingangsdaten zur N-Flächenbilanzierung, bei der erwogen wurde, Teile der Eingangsdaten nicht mehr auf dem bisherigen Weg, sondern direkt über entsprechende Liefervereinbarungen vom Thünen-Institut zu beziehen.
10. Wiederkehrende technische und organisatorische Probleme bei der Bereitstellung der erforderlichen Standardlaufzeitumgebung für Python in der Betriebsumgebung des UBA.

Somit musste sich das Vorhaben zunehmend auf folgende Kernziele fokussieren:

1. Die Neuimplementierung des Rechenkerns zur N-Flächenbilanzierung, d. h. die Nachbildung und Prüfung der Stickstoffflächenbilanzierung, basierend auf neueren Datenbeständen, und die nachweislich korrekte Reproduktion der Rechenwege und Bilanzierungsergebnisse der Referenzimplementierung.
2. Die Analyse und nur teilweise Nachbildung der Vorberechnungen von Eingangsdaten.
3. Die Bereinigung und die Anbringung sicherer, reproduzierbarer Korrekturen struktureller Fehler in den Referenzrechenmappen.

Diese Zielanpassungen erfolgten inkrementell über den Bearbeitungszeitraum 03/2023 – 11/2025.

1 Darstellung des IST-Zustands der Implementierung

Die wissenschaftlichen und konzeptionellen Grundlagen des Modells zur Modellierung von Stoffflüssen sind in Häußermann et al. (2019) beschrieben. Die nachfolgenden Darstellungen beschreiben den Stand der Implementierung. Dieser umfasst

- ▶ die Implementierung der Bilanzierungsvorschrift gem. Häußermann et al. (2019),
- ▶ die Vorverarbeitungsschritte zur Berechnung bzw. Schätzung der notwendigen Eingangsgrößen,
- ▶ die Beschreibung der Implementierung aus technischer Sicht,
- ▶ die Auswertung und Darstellung der bisher vorhandenen Dokumentation zur Implementierung.

1.1 Aufbau und Implementierung

Die aktuelle Implementierung des Modells zur N-Flächenbilanzierung ist vollständig in MS Excel umgesetzt. Eingangsdaten, Datenaufbereitung und die eigentliche Bilanzierung sind auf mehrere Excel-Dateien und darin auf viele Tabellenblätter verteilt. Die aufbereiteten Eingangsdaten, Kontrollparameter und Koeffizienten sind zusammen mit der verformelten Bilanzierung in einer Hauptdatei zusammengefasst.

Die Übertragung der Daten zwischen den einzelnen Dateien, Tabellenblättern und Berechnungsschritten erfolgt teils automatisiert durch Verweisfunktionen und Zellbezüge, teils erfolgt sie auch manuell. Jenseits der wissenschaftlichen Publikationen zur Methodik existiert nur eine marginale Dokumentation zur Implementierung. Eine Entwicklungs- und/oder Anwendungsdokumentation fehlt und das Wissen zur korrekten Bedienung und Weiterentwicklung liegt in den Köpfen der zuständigen Mitarbeitenden. Die Methoden und Rechenschritte zur Aufbereitung der Eingangsdaten sind nicht separat dokumentiert. Das Wissen dazu muss mittels *reverse engineering* aus der vorhandenen Implementierung extrahiert werden.

Die Beschaffung und Verschriftlichung von Wissen zur Funktionsweise des Bilanzierungsmodells, der Datenstrukturen sowie der Rechenwege zur Datenvorverarbeitung und -harmonisierung sind damit eine wesentliche Aufgabe des Projektes.

1.2 Räumliche und zeitliche Granularität

Grundsätzlich bekannt sind die räumliche und zeitliche Granularität der aufbereiteten Eingangsdaten. Die Implementierung rechnet in jahresscharfen Zeitschritten und nutzt als kleinste räumliche Einheit sog. *Regionshöfe*. Diese werden im praktischen Gebrauch auch als *Kreisregionen* bezeichnet und sind aus der aktuellen administrativen Gliederung Deutschlands (auf Kreisebene) abgeleitet. Die Kreisregionen sind in ihrer Anzahl auf 349 begrenzt. Die räumlichen Grenzen der Kreisregionen ergeben sich aus der Zusammenlegung einzelner Kreise und dienen unter anderem der Eliminierung von statistischen Effekten, die sich aus strukturellen Unterschieden zwischen Stadt- und Landkreisen ergeben². Da sich der Zuschnitt einzelner Kreise in der Vergangenheit immer wieder verändert hat, werden die Schritte zur

² Z. B. sind viele Agrarunternehmen in den Stadtkreisen registriert. Deren Agrarflächen befinden sich jedoch in den benachbarten Landkreisen.

Datenaufbereitung im Rahmen der Modellpflege wieder an die administrative Gliederung Deutschlands angepasst.

Im Zuge der Vorprozessierung werden die Eingangsdaten in mehreren Verarbeitungsschritten jahresscharf auf die beschriebenen Kreisregionen – je nach Art und Struktur – aggregiert oder umgerechnet.

Da für die Bilanzierung lückenlose Datenreihen erforderlich sind, werden im Zuge der Vorprozessierung auch sämtliche Datenlücken behoben bzw. befüllt. Das Vorgehen hierzu ist derzeit nicht dokumentiert, d. h. die angewendeten Verfahren müssen aus den implementierten Formeln rückwirkend herausgelesen und neu beschrieben werden.

1.3 Statistisches Mengengerüst und Eingangsdaten

Im Zuge der Vorbereitung einer Übernahme des N-Bilanzierungsmodells durch das UBA wurden in jüngerer Vergangenheit die einfließenden Datenquellen systematischer dokumentiert. Die Gesamtheit dieser einfließenden Daten, die im Wesentlichen der Wirtschafts- und Agrarstatistik auf Bundes- und Landesebene entstammen, wird gemeinhin als **statistisches Mengengerüst** bezeichnet.

Die Erstellung einer Übersicht zu diesen Eingangsdaten, die auch den exakten Stand der Implementierung darstellen, ist innerhalb der ersten Projektmonate nicht möglich gewesen. Die Erarbeitung einer entsprechenden Dokumentation im Rahmen des angesprochenen *reverse engineering*s war Gegenstand der Projektbearbeitung. Ausgangspunkt war der Entwicklungsstand der Rechenmappen Stand Juli/2023 auf dessen Grundlage die Neuimplementierung des Bilanzierungsmodells erfolgte.

1.4 Modellpflege und Weiterentwicklung

Pflege und Weiterentwicklung des N-Bilanzierungsmodells fanden in der Vergangenheit etwa im Turnus von zwei bis drei Jahren auf der Basis von Forschungsaktivitäten statt. Neben allgemeinen Aufgaben zum Stickstoffmonitoring werden in diesem Zuge auch anlassbezogene Modellrechnungen durchgeführt, die die Auswirkung von regulatorischen Maßnahmen auf die Stickstoffeinträge aus der Landwirtschaft simulieren.

Bis zum Jahr 2025 wurde das Modell noch von der Arbeitsgruppe um Dr. Martin Bach an der JLU Gießen betreut und die Zeitreihe bis zum Jahr 2023 fortgeschrieben. Bisher hat das UBA für eigene Aktivitäten im Stickstoffmonitoring und der Politikberatung unter anderem auf Dienstleistungen dieser Arbeitsgruppe zurückgegriffen. Da frühzeitig bekannt war, dass sich die Arbeitsgruppe spätestens 2025 auflösen wird, sollten Betrieb, Pflege und Weiterentwicklung ans UBA übergehen.

2 Definition des SOLL-Zustandes

2.1 Szenarien, Modelllauf und Basislauf

Modelle sind zweckmäßige Abstraktion der Realität und dienen im Falle der N-Flächenbilanzierung der näherungsweise Abschätzung von Stickstoffeinträgen und gasförmigen Verlusten (Emissionen) auf und von landwirtschaftlichen Flächen. Zur Berechnung dieser Kenngrößen sind vielfältige Eingangsdaten und Annahmen erforderlich, die im Vorfeld beschafft, aufbereitet und an die jeweilige Aufgabenstellung angepasst werden müssen. Typische Aufgaben in der N-Flächenbilanzierung zerfallen in zwei Gruppen:

1. Die Bilanzierung tatsächlicher Stickstoffflüsse auf Agrarflächen auf Basis der tatsächlichen Rahmenbedingungen (*status quo*)
2. Die Bilanzierung von Stickstoffflüssen auf Agrarflächen unter angenommenen veränderten Rahmenbedingungen

Um beide Gruppen sprachlich besser zu unterscheiden, haben sich die Begriffe *Szenario* und *Basislauf* eingebürgert. Ein *Szenario* umfasst statistisch gesicherte Größen sowie Annahmen zur wirtschaftlichen Tätigkeit der Agrarbetriebe, zu Bewirtschaftungsformen und Stoffeinsätzen in der Agrarwirtschaft, zum regulatorischen Rahmen sowie zu physikalisch-chemischen Speicher- und Umwandlungsprozessen im Boden und Atmosphäre.

Ein einzelnes **Szenario** umfasst ein bestimmtes konsistentes Bündel von Annahmen zum *status quo* bzw. zu veränderten Rahmenbedingungen. Das Modell zur N-Flächenbilanzierung erzeugt in einem Durchlauf (= **Modelllauf** oder auch Realisierung) die N-Flächenbilanzen für genau ein Szenario. Für einen Modelllauf werden mittels statistischer Mengengerüste und einstellbarer Modellparameter, die *gesicherte Größen* und die *Annahmen* des Szenarios in geeigneter Form quantifiziert. Die Berechnung mehrerer Szenarien erfordert entsprechend viele Modellläufe.

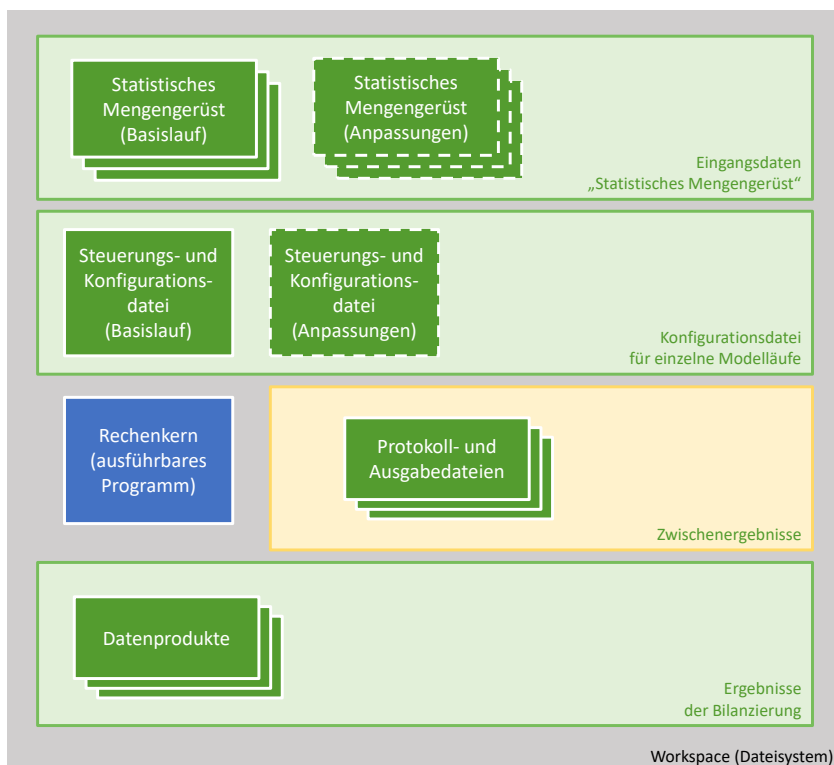
Der **Basislauf** ist ein Modelllauf, der ein Szenario mit herausgehobener Bedeutung realisiert. Es ist in der Regel das Szenario, dessen Annahmen sich – unter Berücksichtigung des Zweckes und der Grenzen des Modells – dem *status quo* in einem zurückliegenden Auswertzeitraum bestmöglich annähern.

Aufgabe des *Basislaufes für die N-Flächenbilanzierung* ist die bestmögliche Abschätzung von Stickstoffbilanzen für die vergangenen Jahre, rückwirkend bis 1995.

2.2 Komponenten des Bilanzierungsmodells

Für die Diskussion der Bestandteile des Bilanzierungsmodells ist es zweckmäßig auf das Konzept von *Workspaces* zurückzugreifen (Abbildung 1). Ein Workspace umfasst dabei alle Komponenten (Daten und ausführbaren Code), die zur Ausführung eines Modell-Laufs notwendig sind.

Abbildung 1: Struktureller Aufbau eines Workspaces für das Bilanzierungsmodell



Quelle: eigene Darstellung, PIKOBAYTES GmbH

Die Komponenten eines Workspace werden nachfolgend kurz vorgestellt.

Statistisches Mengengerüst

Das statistische Mengengerüst umfasst sämtliche Eingangsdaten, die für den Modellantrieb notwendig sind. Hierzu gehören z. B. Daten zu Art und Größe landwirtschaftlicher Flächen, landwirtschaftliche Ertragsstatistiken oder statistische bzw. geschätzte Daten zum Einsatz von Düngemitteln. Jeder Datenpunkt im statistischen Mengengerüst ist üblicherweise mit einer Zeitangabe versehen und die Daten liegen jährlich aufgelöst vor.

Steuerungs- und Konfigurationsdatei

Neben den Daten aus dem statistischen Mengengerüst, die das Modell antreiben, sind für die Ausführung der Berechnungs- und Bilanzierungsvorschriften bestimmte Parameter und Koeffizienten erforderlich. Diese sind bis auf Ausnahmen zeitunabhängige Konstanten, die für einen Modellauf bzw. ein Szenario geeignet definiert werden müssen. Die *Steuerungs- und Konfigurationsdatei* definiert, wie die Eingangsdaten aus dem *statistischen Mengengerüst* vom *Rechenkern* verarbeitet werden.

Rechenkern

Der Rechenkern enthält den ausführbaren Programmcode des Bilanzierungsmodells. Er erzeugt anhand der Angaben in der Steuerungs- und Konfigurationsdatei aus den Daten des *statistischen Mengengerüsts* die Bilanzierungsergebnisse sowie zusätzliche Protokoll- und Ausgabedateien.

Protokoll- und Ausgabedateien

Im Zuge der Datenverarbeitung erzeugt der Rechenkern während seiner Ausführung unterschiedliche Protokoll- und Ausgabedateien, die der Überwachung und Kontrolle des

Programmablaufes dienen. *Protokolldateien* (auch Log-Dateien) enthalten vorwiegend technische Informationen, die vom Rechenkern während der Ausführung fortlaufend erzeugt werden. *Ausgabedateien* beinhalten wichtige Zwischenergebnisse der Berechnung, die im Wesentlichen für Kontrollzwecke ausgegeben werden. In bestimmten Fällen können diese Zwischenergebnisse auch für weiterführende Datenauswertungen herangezogen werden.

Datenprodukte (der N-Flächenbilanzierung)

Der Rechenkern produziert am Ende eines Durchlaufs Datenprodukte in definierter Qualität und Struktur, die räumlich und zeitlich aufgelöste Ergebnisse zur N-Flächenbilanzierung enthalten. Diese Datenprodukte sind für die Weiterverarbeitung in den Fachabteilungen sowie zur Weitergabe an Dritte bestimmt und dienen der Erfüllung von Berichtspflichten und Aufgaben der Politikberatung sowie zur Information der Öffentlichkeit.

Steuerungs- und Konfigurationsdatei (Anpassungen)

Für die Berechnung von Szenarien und Varianten muss das Bilanzierungsmodell angepasst werden. In einfachen Fällen ist dafür lediglich die Änderung von Parametern und Koeffizienten in der Steuerungs- und Konfigurationsdatei erforderlich. Anpassungen gegenüber den Standardeinstellungen werden separat konfiguriert und „überschreiben“ zu Laufzeit die jeweiligen Standardwerte.

Statistisches Mengengerüst (Anpassungen)

In bestimmten Fällen ist eine Anpassung von Parametern und Koeffizienten in der Steuerungs- und Konfigurationsdatei für die Berechnung von Szenarien und Varianten nicht ausreichend. In diesen Fällen sind weitergehende Anpassungen einzelner Eingangsdaten aus dem statistischen Mengengerüst erforderlich. Diese werden separat von den Standardwerten des statistischen Mengengerüsts abgelegt und „überschreiben“ zu Laufzeit die jeweiligen Standardwerte.

2.3 Modellpflege, Versionen und Versionierung des Bilanzierungsmodells

Wie bereits in Kapitel 2.1 beschrieben, kann das Modell zur N-Flächenbilanzierung die Stickstoffbilanzierung für unterschiedliche Szenarien berechnen. Die Realisierung des Basislaufs steht im Vordergrund da dieser für das Monitoring der Stickstoffeinträge benötigt wird. Dieser Basislauf wird derzeit i. d. R. jährlich aktualisiert und auf das aktuell verfügbare statistische Mengengerüst angepasst. Bei erforderlichen strukturellen oder inhaltlichen Änderungen der Eingangsdaten (Zusammenlegung von Kreisen, veränderte Datenquellen, veränderte Statistikdaten in abweichender Granularität) wird das Modell entsprechend angepasst, sodass auch alle zurückliegenden Jahre mit den veränderten Daten neu gerechnet werden können.

Dieses Vorgehen entspricht der gelebten Praxis im IST-Zustand und soll bis auf Weiteres so fortgeführt werden.

Für diese turnusmäßig aktualisierten Basisläufe ist es zwingend erforderlich, alte Versionen so zu archivieren, dass (1) Vergleichsrechnungen ermöglicht werden und (2) Änderungen am Modellkern und den Zwischenergebnissen nachvollziehbar bleiben.

Weiterhin werden für besondere Aufgaben, Projekte und Datengesuche zusätzliche Szenarien berechnet. Diese sind vom Basislauf abgeleitet und passen Mengengerüste bzw. Steuerungs- und Konfigurationsparameter an die Vorgaben des jeweiligen Szenarios an. In Ausnahmefällen muss davon ausgegangen werden, dass auch kleinere Anpassungen am Modellkern vorgenommen werden.

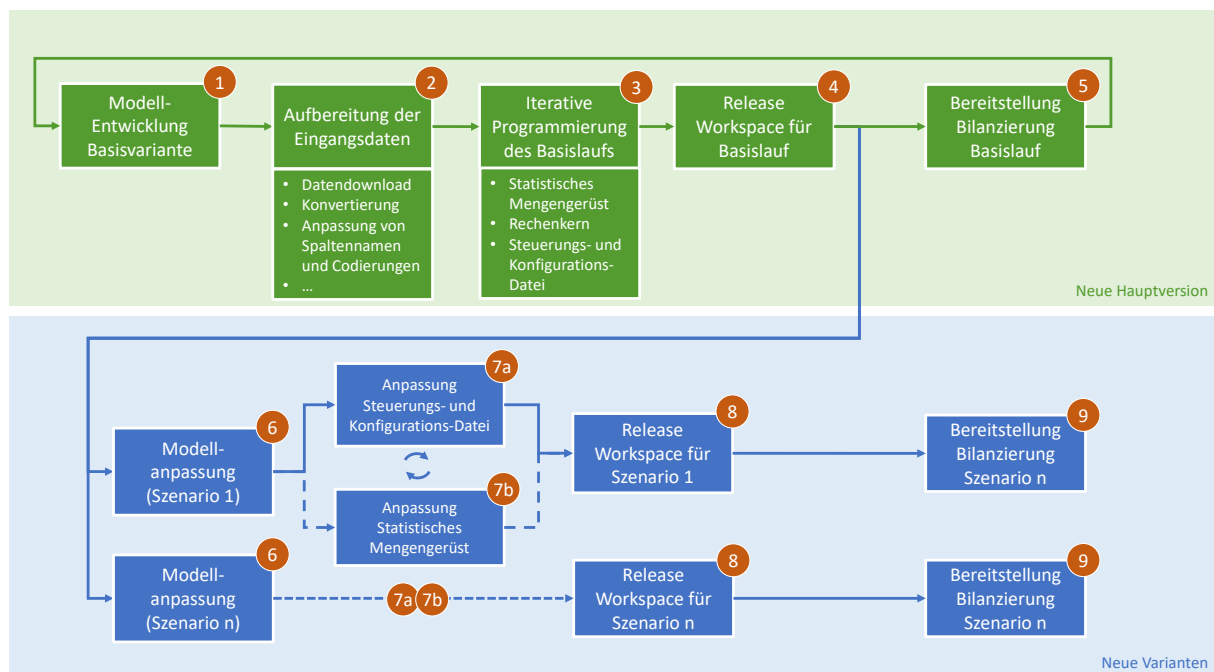
Für diese bedarfsgesteuert realisierten Szenarien ist es zwingend erforderlich, (1) einen Bezug zum übergeordneten Basislauf herzustellen, (2) auch im Nachhinein noch Vergleichsrechnungen vorzunehmen und (3) Änderungen an statistischem Mengengerüst, Parametrierung, Zwischenergebnissen und Modellkern nachvollziehbar zu dokumentieren.

Im Falle nachträglich festgestellter Berechnungs- oder Implementierungsfehler muss sowohl für Basisläufe als auch die Szenarien nachvollziehbar sein, in welche Versionen und Datenbereitstellungen sich Fehler ggf. fortgepflanzt haben und ob ein Fehler so groß ist, dass bereits stattgefundenene Datenlieferungen mit korrigierten Daten ersetzt werden müssen.

2.4 Konzept zur Modellversionierung

Basierend auf den in Kapitel 2.3 beschriebenen Rahmenbedingungen und Anforderungen zeigt Abbildung 2 schematisch ein Vorgehen zur Modellversionierung.

Abbildung 2: Ablaufschema zur Erstellung neuer Hauptversionen und abgeleiteter Varianten



Quelle: eigene Darstellung, PIKobytes GmbH

Die einzelnen Schritte und Artefakte zur Entwicklung neuer Hauptversionen sowie abgeleiteter Modellvarianten werden nachfolgend kurz beschrieben.

(1) Modellentwicklung Basisvariante

Die turnusmäßige Entwicklung einer neuen Basisvariante des N-Bilanzierungsmodells beginnt. Die Entwicklung der neuen Basisvariante baut in der Regel auf der vorangegangenen Basisvariante auf.

(2) Aufbereitung der Eingangsdaten

Die Beschaffung der Eingangsdaten ist ein manueller Prozess, der aufgrund der Evolution der Datenquellen und der Schnittstellen der datenhaltenden Stellen auch nicht wirtschaftlich zu automatisieren ist. Die Datenaufbereitung umfasst Tätigkeiten, wie den Download und die Sichtung der neuen Daten, die Konvertierung von unterschiedlichen Dateiformaten, die Anpassung von Codierungen und Spaltennamen sowie viele andere kleinteilige,

quellenspezifische Tätigkeiten zur Datenumformung. Mit Abschluss dieses Arbeitsschrittes liegen aufbereitete, homogen strukturierte Eingangsdaten vor, die für die nachfolgende Programmierung erforderlich sind.

(3) Iterative Programmierung des Basislaufs

Der neue Basislauf wird iterativ entwickelt. Der Bestand an Eingangsdaten aus (2) wird strukturell aufbereitet und mit verschiedenen Verfahren zur Umschätzung und Lückenbefüllung in ein statistisches Mengengerüst überführt. Die Programmierung des Rechenkerns wird den neuen Bedürfnissen angepasst. Nach mehreren Entwicklungszyklen ist der Basislauf getestet und anhand der Zwischenergebnisse und der erzeugten Datenprodukte validiert.

(4) Release-Workspace für den Basislauf

Für den geprüften Basislauf wird ein neues Workspace-Release angelegt.

(5) Bereitstellung Bilanzierung gem. Basislauf

Auf Basis von (4) wird eine neue N-Flächenbilanzierung bereitgestellt.

(6) Modellanpassung (Szenario)

Für die Berechnung eines Szenarios wird eine Modellanpassung erforderlich. Diese zweigt zunächst vom Release-Workspace des Basislaufs ab.

(7) Anpassung von Datenbasis und Modellkonfiguration

Für das Szenario sind Anpassungen an der Steuerungs- und Konfigurationsdatei (7a) und/oder des statistischen Mengengerüsts (7b) erforderlich.³

Die Anpassungen werden iterativ vorgenommen, geprüft und getestet.

(7a) Anpassung der Steuerungs- und Konfigurationsdatei

Für die Realisierung des Szenarios ist eine Anpassung der Steuerungs- und Konfigurationsdatei notwendig. Die Anpassungen werden auf Grundlage der Steuerungs- und Konfigurationsdatei des Basislaufes vorgenommen.

(7b) Anpassung des statistischen Mengengerüsts

Für die Realisierung des Szenarios ist eine Anpassung des statistischen Mengengerüsts notwendig. Die Anpassungen werden auf Grundlage des statistischen Mengengerüsts des Basislaufes vorgenommen.

(8) Release-Workspace für das Szenario

Für die geprüfte Realisierung des Szenarios wird ein neues Workspace-Release (als Variante des Basislaufs) angelegt.

(9) Bereitstellung Bilanzierung gem. Szenario

Auf Basis von (8) wird eine neue N-Flächenbilanzierung für das Szenario bereitgestellt.

2.5 Mengengerüst

Das Mengengerüst für die N-Flächenbilanzierung umfasst Angaben zu Anzahl und Größe der Einzeldatensätze des statistischen Mengengerüsts, der Zahl der Steuerungs- und

³ In seltenen Fällen können auch Anpassungen am Programmcode erforderlich werden. Dieser Fall wird im Versionierungskonzept antizipiert, aber in der Darstellung aufgrund seiner Seltenheit und aus Übersichtsgründen nicht explizit ausformuliert.

Konfigurationsparameter sowie zu Anzahl, Häufigkeit und Datenvolumen von Basisläufen und Szenarien. Diese Angaben finden Eingang in die Systemkonzeption und die Darstellung einer effektiven, wartbaren und wirtschaftlichen Betriebsumgebung. Die entsprechenden Kenngrößen sind in Tabelle 1 aufgeführt.

Tabelle 1: Mengengerüst zur N-Flächenbilanzierung (Abschätzung)

Kenngröße	Menge
Statistisches Mengengerüst	1 Tabelle je Kalenderjahr von 1995 bis zum jeweiligen Berichtsjahr (für Berichtsjahr 2023: 29 CSV-Dateien, ca. 5,6 MB)
Anzahl der Steuerungs- und Konfigurationsparameter	ca. 271 (davon 93 Einzelwerte und 178 Tabellen), ca. 5 MB
Protokoll- und Ausgabedateien	Pro Berechnungslauf <ul style="list-style-type: none"> • 1 Logdatei • je Ergebnisgröße: ca. 153 Einzeltabellen (für Berichtsjahr 2021 ca. 23 MB)
Datenprodukte zur N-Flächenbilanzierung	Pro Berechnungslauf <ul style="list-style-type: none"> • je Kalenderjahr: 5 Einzeltabellen, jeweils ca. 10 MB
Fortschreibung des Basislaufes	Jährlich
Anzahl neuer Szenarien / Varianten	6 Varianten / Jahr
Jährliches Datenaufkommen	ca. 1000 Einzeltabellen, 1 GB

Gemessen an den Fähigkeiten zeitgemäßer Arbeitsplatzrechner und Server ist das abgeschätzte Datenaufkommen eher gering. Eine besondere Hardwareausstattung für die Datenspeicherung ist nicht erforderlich.

2.6 Systemkonzeption

Für die Systemkonzeption werden in diesem Kapitel unterschiedliche Realisierungsvarianten für Datenhaltung, Rechenkern und Versionierung diskutiert und Vorzugsvarianten begründet abgeleitet.

2.6.1 Datenhaltung

2.6.1.1 Anforderungen und Randbedingungen

Die Datenhaltung für die Eingangsdaten und Datenausgaben soll möglichst einfach zugänglich und gleichzeitig performant realisiert werden. Da das statistische Mengengerüst tabellarisch strukturiert ist, werden Daten aller Wahrscheinlichkeit nach mit Standardwerkzeugen (Texteditoren, Tabellenkalkulation) erzeugt, gelesen und editiert. Ein textbasiertes Speicherformat ist damit von Vorteil.

Im Hinblick auf das Workspace-Konzept und die Modellversionierung ist eine dateibasierte Datenhaltung von Vorteil.

2.6.1.2 Vorzugsvariante

Als allgemein gängiges textbasiertes Dateiformat wird der Einsatz von CSV empfohlen. CSV-Dateien sind mit allen gängigen Texteditoren und Tabellenkalkulationen lesbar und schreibbar. Für die programmatische Verarbeitung stehen in fast allen Programmiersprachen Standardbibliotheken für die Ein- und Ausgabe zur Verfügung.

2.6.1.3 Alternativen

Eine mögliche Alternative zu CSV ist das ebenfalls weit verbreitete text-basierte Format JSON. Da JSON jedoch vorwiegend für hierarchisch strukturierte Daten konzipiert und für tabellarische organisierte Daten schlecht geeignet ist, entstünden dort eher Nachteile. Für die Ablage der Steuerungs- und Konfigurationsparameter kann eine hierarchische Strukturierung ggf. Vorteile bringen. Für diesen speziellen Zweck wurde der Einsatz von JSON im Zuge der Umsetzung geprüft.

Prinzipiell wäre auch eine Datenhaltung in einer relationalen Datenbank, wie z.B. PostgreSQL möglich. Datenbanken bieten Vorteile in Multi-User-Umgebungen und können durch Schema-Definitionen eine hohe Datenintegrität gewährleisten.

Die Notwendigkeit der Definition eines relationalen Schemas ist für bestimmte Aufgaben allerdings auch von Nachteil. Im Zuge der turnusmäßigen Fortschreibung haben sich in der Vergangenheit große Teile des Datenbestandes strukturell und inhaltlich verändert. Hier steht eine starre relationale Datenbank einer effizienten Weiterentwicklung und einer Erzeugung von Modellvarianten eher im Weg.

Weiterhin wird der Vorgang der Versionierung komplizierter, da Rechenkern und Daten nicht mehr im gleichen Workspace geführt werden können.

2.6.2 Programmierung des Rechenkerns

2.6.2.1 Anforderungen und Randbedingungen

Die Programmierung des Rechenkerns sollte in einer gängigen und möglichst weit verbreiteten Programmier- oder Skriptingsprache erfolgen, die für die Umsetzung ähnlicher Aufgaben bereits praktisch bewährt ist. Bei der Auswahl sollte antizipiert werden, dass technisch erfahrene Mitarbeitende des UBA potenziell selbst Anpassungen und Entwicklungen am Modellkern vornehmen können. Eine Sprache mit einer geringen Eintrittshürde und weiter Verbreitung ist daher zu bevorzugen.

Für die gewählte Sprache sollte weiterhin eine frei nutzbare Ausführungsumgebung zur Verfügung stehen, die eine ausreichend hohe Verarbeitungsgeschwindigkeit sicherstellt.

2.6.2.2 Vorzugsvariante

Die Implementierung erfolgt vorzugsweise in der Skriptsprache Python. Die Sprache ist weit verbreitet und wird auch im Bereich des wissenschaftlichen Rechnens häufig eingesetzt. Für Python stehen weiterhin optimierte Module bzw. Pakete für numerische Berechnungen bereit, sodass eine ausreichend hohe Verarbeitungsgeschwindigkeit sichergestellt ist.

Die Ausführungsumgebung ist frei nutzbar und steht für gängige Betriebsumgebungen bereit. Eine Kompilierung des Programmcodes ist nicht erforderlich, die Skripte sind direkt ausführbar und testbar.

Python bietet darüber hinaus die Möglichkeit, den Rechenkern künftig in Webdienste einzubinden.

2.6.2.3 Alternativen

Die Sprachen R und Java waren mögliche Alternativen zur Umsetzung des Rechenkerns und sind ebenfalls sehr weit verbreitet.

Die Typsicherheit von Java bietet gewisse Vorteile bei der Vermeidung von Programmierfehlern, die Sprache ist jedoch für die beschriebene Aufgabe eher selten im Einsatz.

Die Sprache R wird für wissenschaftliches Rechnen, anspruchsvolle statistische Auswertungen sowie die Erstellung von Diagrammen und Grafiken eingesetzt. Vom Funktionsumfang ist die Sprache ähnlich gut geeignet wie Python, die Ausführungsgeschwindigkeit ist jedoch geringer. Vom Ansatz her ist R für die interaktive Analyse konzipiert. Die Einbindung eines R-basierten Rechenkerns in einen Webdienst ist – insbesondere unter Sicherheitsaspekten – eher unüblich.

2.6.3 Versionierung

2.6.3.1 Anforderungen und Randbedingungen

Die funktionalen Anforderungen an die Versionierung sind in den Kapiteln 2.3 und 2.4 benannt. Aus Gründen der Konsistenz und Übersichtlichkeit ist eine Versionierung von Workspaces (Code und Daten, inklusive der Rechenergebnisse) von großem Vorteil. Im Rahmen des angesetzten Mengengerüsts (vgl. Kapitel 2.5) bleibt die Anzahl der Versionen insgesamt überschaubar.

2.6.3.2 Vorzugsvariante

Die Modellversionierung erfolgt dateibasiert über Workspaces. In Anbetracht des Mengengerüsts und der verwendeten Datenformate ist die **Nutzung eines Versionskontrollsystems** zu empfehlen. Dieses kann entweder selbst gehostet am UBA betrieben oder als Cloud-Lösung eingekauft werden.

Versionskontrollsysteme verfügen über grafische Werkzeuge zur übersichtlichen Darstellung von Entwicklungszweigen und Änderungen zwischen unterschiedlichen Entwicklungsständen. Sie sind zudem als Mehrbenutzer*innenumgebungen ausgelegt und ermöglichen neben der Pflege zentraler Modellversionen (in Form von *Branches*) auch die Pflege interner Entwicklungszweige (in Form von *Forks*), die ab dem Zeitpunkt der Verzweigung unabhängig weitergeführt werden können oder später ggf. in das Hauptprojekt wieder integriert werden können.

2.6.3.3 Varianten zur Umsetzung

Für die Versionierung der Workspace gibt es zwei Möglichkeiten: (1) verzeichnisbasiert und ohne Versionskontrollsystem und (2) in einem Versionskontrollsystem.

In der **verzeichnisbasierten Umsetzung** werden Hauptversionen und Varianten in eigenen Ordnern geführt. Die Ordner sind dabei sinnvoll zu benennen (z.B. 2023-Basislauf, 2023-SzenarioXX o. ä.). Im Falle von nachträglichen Änderungen und Fehlerbehebungen ist es zweckmäßig, dafür eigene Subversionen zu pflegen (z.B. 2023-Basislauf-1.0, 2023-Basislauf-1.1 usw.).

Der Einsatz eines **Versionskontrollsystems** (wie z. B. GIT) bietet gegenüber der rein verzeichnisbasierten Variante eine „echte“ Versionierung. Anstelle der händischen Kuratierung von Versionsverzeichnissen definieren Versionskontrollsysteme einen Prozess für das Erzeugen

neuer Versionen und die Protokollierung von Änderungen.⁴ Zusätzliche Vorteile von Versionskontrollsystemen sind die bessere Abbildung von Fehlerbehebungen / Anpassungen, der Darstellung von Abhängigkeiten zwischen den Versionen, der Möglichkeit, Änderungen zu kommentieren und zu visualisieren sowie häufig eine effizientere Datenspeicherung durch Deduplikation gleicher Daten zwischen verschiedenen Varianten/Versionen.

Sowohl in der **verzeichnisbasierten Umsetzung** als auch in der Variante mit **Versionskontrollsystem** ist die Berechnung sogenannter *Diffs* möglich. Diese geben detailliert Auskunft darüber, an welchen Stellen sich zwei Versionen unterscheiden. Die Einstiegshürde für unerfahrene Nutzer*innen ist in der verzeichnisbasierten Umsetzung deutlich geringer. Versionskontrollsysteme erfordern von den Nutzer*innen eine Einarbeitung in Versionierungskonzepte und entsprechende Werkzeuge, bieten dafür aber den Vorteil eines deutlich stärker vorgegebenen Versionierungsprozesses. Im Zuge der Umsetzung konnte aufgrund vorliegender Einschränkungen am UBA noch kein Versionierungskonzept mittels Versionskontrollsystem eingerichtet werden. Dies ist zukünftig entsprechend vorbereitet und zukünftig vorgesehen, sobald die Möglichkeit der Nutzung eines Versionskontrollsystems für die Fachanwendenden am UBA geschaffen wurde.

⁴ Diese Verfahren funktionieren i. d. R. auf Textdateien, was auf Grundlage der Konzeption von Datenhaltung und Programmierung des Rechenkerns in den dargestellten Vorzugsvarianten aber immer der Fall ist.

3 Durchführung und Umsetzung

3.1 Allgemeines Vorgehen

Die Neuimplementierung des Stickstoffbilanzierungsmodells wurde in enger Anlehnung an ein agiles Vorgehensmodell durchgeführt. Dieses Vorgehensmodell hat sich insbesondere bei der Umsetzung von Softwareprojekten etabliert, die sehr komplex, bei denen die Anforderungen und Lösungswege nicht von vornherein klar sind und die eine häufige und tiefe inhaltliche Abstimmung mit den Projektpartner*innen erfordern.

Bereits während der Analyse des IST-Zustandes wurde erkennbar, dass die Neuimplementierung des Bilanzierungsmodells und der vorgelagerten Rechenmappen und Datenaufbereitungsschritte in erheblichem Maße *reverse engineering* erfordert und dass große Teile der Dokumentation zur bestehenden Implementierung im Zuge eines agilen Vorgehens zunächst erst noch erstellt werden müssen.

Die einzelnen Arbeitsschritte und Projektartefakte werden nachfolgend dargestellt.

3.2 Bereitstellung einer Referenzimplementierung des Bilanzierungsmodells

3.2.1 N Referenzimplementierung 07/2023

Ausgangspunkt der Entwicklung war die Bereitstellung einer Referenzimplementierung mit Bearbeitungsstand 07/2023, die ein lauffähiges Bilanzierungsmodell für die Kalenderjahre 1995 bis 2021 in Form von mehreren Excel-Rechenmappen sowie assoziierten Rechenmappen zur Datenvorverarbeitung enthält. Die Struktur der Hauptrechenmappe ist in Abbildung 3 dargestellt.

In dieser Referenzimplementierung sind sämtliche Rechenschritte zur Bilanzierung in Excel verformelt. Durch sehr hohe Komplexität der Rechenmappen war eine automatisierte Ausführung der Berechnungen nicht möglich. Die Berechnungen in den einzelnen Tabellenblättern mussten manuell und nach einem festen Vorgehen ausgelöst werden, um in mehreren Schritten die Bilanzierungsergebnisse aus den Eingangsdaten zu berechnen. Die Anforderungen an die Ausführungsumgebung sind vergleichsweise hoch – ein stabiler Betrieb der Rechenmappen erfordert einen leistungsfähigen Arbeitsplatzrechner mit 32 GB Arbeitsspeicher.

Die Referenzimplementierung stellt zwei wichtige Analyse- und Prüfsichten für die Neukonzeption des Rechenkerns zur Verfügung:

- ▶ Numerische Eingangsdaten, Zwischen- und Endergebnisse in der Stickstoffbilanzierung
- ▶ Formelmäßige Beschreibungen zur schrittweisen Ableitung der Bilanzgrößen aus den (vorprozessierten) Eingangsdaten

Mit diesen Informationen lagen prinzipiell alle benötigten Informationen für den Nachbau und die anschließende Funktionsprüfung des Rechenkerns vor. Allerdings ließen sich die Rechenformeln nicht direkt in eine andere Ausführungsumgebung übertragen. Die genutzten Formeln und deren Struktur sind eng an die Fähigkeiten und Eigenschaften von Tabellenkalkulationsprogrammen (insbesondere Excel) geknüpft und haben keine direkte

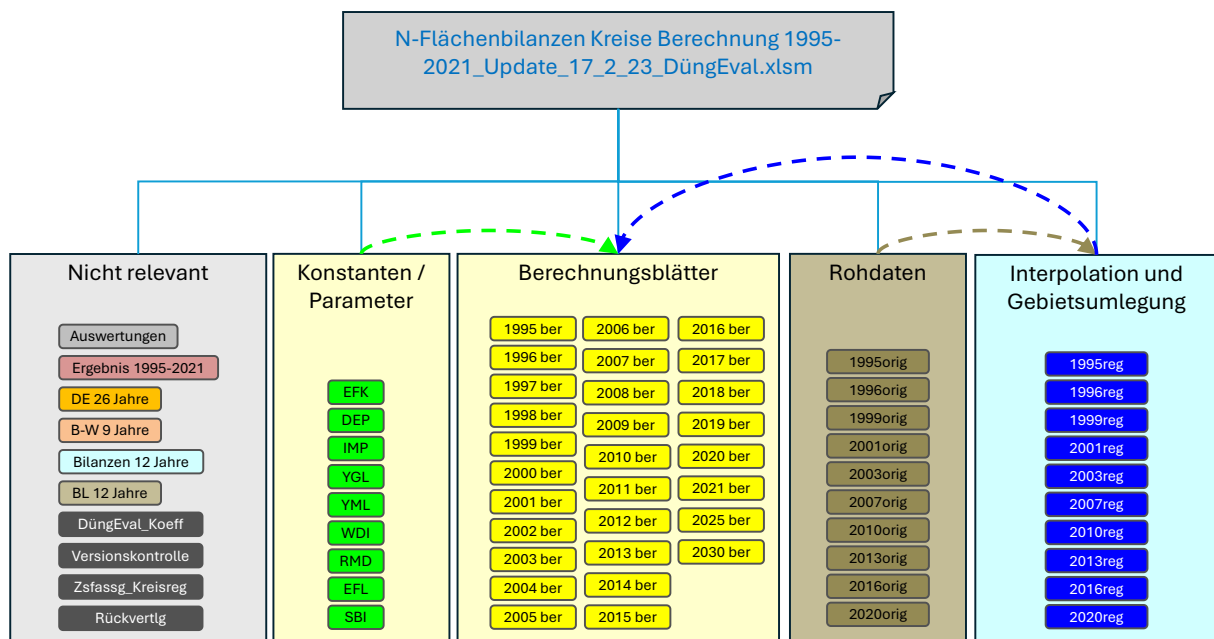
Entsprechung in gängigen Programmiersprachen. Sie dienen daher als Ausgangspunkt für das *reverse engineering* der darunter liegenden Rechenvorschriften.

Die Verknüpfung der Rechenmappen untereinander erfolgt in Teilen manuell durch das Umkopieren von Daten zwischen einzelnen Tabellenblättern. Dieser Prozess ist in der Referenzimplementierung nicht abgebildet und musste durch Interviews mit den bisherigen Entwicklern des Bilanzierungsmodells erarbeitet und dokumentiert werden.

Die Referenzimplementierung enthält folgende 68 Tabellenblätter (hier alphabetisch sortiert):

'1995 ber', '1995orig', '1995reg', '1996 ber', '1996orig', '1996reg', '1997 ber', '1998 ber', '1999 ber', '1999orig', '1999reg', '2000 ber', '2001 ber', '2001orig', '2001reg', '2002 ber', '2003 ber', '2003orig', '2003reg', '2004 ber', '2005 ber', '2006 ber', '2007 ber', '2007orig', '2007reg', '2008 ber', '2009 ber', '2010 ber', '2010orig', '2010reg', '2011 ber', '2012 ber', '2013 ber', '2013orig', '2013reg', '2014 ber', '2015 ber', '2016orig', '2016reg', '2017 ber', '2018 ber', '2019 ber', '2020 ber', '2020orig', '2020reg', '2021 ber', '2025 ber', '2030 ber', 'Auswertungen', 'B-W 9 Jahre', 'BL 12 Jahre', 'Bilanzen 12 Jahre', 'DE 26 Jahre', 'DEP', 'DüngEval_Koeff', 'EFK', 'EFL', 'Ergebnis 1995-2021', 'IMP', 'RMD', 'Rückvertlg', 'SBI', 'Versionskontrolle', 'WDI', 'YGL', 'YML', 'Zusfassg_Kreisreg'.

Abbildung 3: Struktur der Hauptrechenmappe



Die Abbildung zeigt eine schematische Darstellung der Tabellenblätter in der zentralen Excel-Datei der Stickstoffflächenbilanz, eingeteilt in Kategorien. Die gestrichelten Pfeile zeigen welche Bezüge es zwischen den Tabellenblättern der Kategorien gibt.

Quelle: eigene Darstellung, PIKobytes GmbH

Von diesen Tabellenblättern wurden die in Tabelle 2 genannten als relevant für die Portierung der N-Flächenbilanz eingestuft.

Tabelle 2: Relevante Tabellenblätter in der Referenzimplementierung

Kategorie	Namen
Berechnungsblätter mit der Kernberechnung	„1995 ber“ bis „2021 ber“ sowie „2025 ber“ und „2030 ber“
Tabellenblätter mit Konstanten bzw. statischen Variablen	"EFK", "RMD", "YML", "EFL", "YGL", "IMP", "SBI", "DEP", "WDI"
Rohdaten aus Officialstatistik, die alle zwei bis vier Jahre vorliegen	"1995orig", "1996 orig", "1999orig", "2001orig", "2003orig", "2007orig", "2010orig", "2013orig", "2016orig", "2020orig"
Interpolation und Gebietsumlegung der Originaldaten	"1995reg", "1996 reg", "1999reg", "2001reg", "2003reg", "2007reg", "2010reg", "2013reg", "2016reg", "2020reg"

Die Originaldatenblätter und die Regionalblätter enthalten Vorberechnungen, deren Ergebnisse in den Berechnungsblättern verwendet werden (teilweise ohne Verknüpfung). Da das Augenmerk zunächst auf der Umsetzung der Kernberechnung lag, wurden diese Blätter nach einer anfänglichen Analyse nicht weiter betrachtet.

Innerhalb der Referenzimplementierung wurde zu Projektbeginn auf Rat der AG Bach das Berechnungsblatt '2019 ber' als Referenz für die Kernberechnung festgelegt, da sich einige Formeln im Lauf der Jahre geändert haben. Grundsätzlich sollten die Formeln über alle Berechnungsblätter hinweg bis auf wenige bekannte Ausnahmen konsistent sein.

Abbildung 4: Tabellenblatt 2019 ber

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2		Jahr/Maß	2019													
3		Rückfall	2019													
4																
5				Zusammenfassung												
6				Gebietseinheit Landwirtschaftliche genutzte Fläche (LF)								Anbau Ackerland				
7		Bezeichnung der Regionaleinheit		Insgesamt	Ackerland (insgesamt)	Dauerkulturen	Dauergrünland		Rebland	Übrige Dauerkulturen	Ackerland	Getreide				
8									darunter ertragsarmes	berechnet Diff. Dauerkulturen gegen Mittel	(insgesamt)	(insgesamt)	Weizen	Roggen und Winterweizen		
9		171-01-4		D. BL. FS 3 R 3.2.1. K. Extrapolation ASE 2016	D. BL. FS 3 R 3.2.1. K. Extrapolation ASE 2016	D. BL. FS 3 R 3.2.1. K. Extrapolation ASE 2016	D. BL. FS 3 R 3.2.1. K. Extrapolation ASE 2016	D. BL. FS 3 R 3.1.2. K. geschätzt	D. BL. FS 3 R 3.2.1. K. Extrapolation ASE 2016	D. BL. FS 3 R 3.2.1. K. Extrapolation ASE 2016	D. BL. FS 3 R 3.2.1. K. Extrapolation ASE 2016	D. BL. FS 3 R 3.2.1. K. Extrapolation ASE 2016	D. BL. FS 3 R 3.2.1. K. Extrapolation ASE 2016	D. BL. FS 3 R 3.2.1. K. Extrapolation ASE 2016	D. BL. FS 3 R 3.2.1. K. Extrapolation ASE 2016	D. BL. FS 3 R 3.2.1. K. Extrapolation ASE 2016
10		Name	Anzahl	ha	ha	ha	ha	ha	ha	ha	ha	ha	ha	ha	ha	ha
11		Zuordnung	Reg. Einheit													
13	DG	DG	Deutschland	400	Kontrollgr.	16.666.000	11.713.700	199.500	4.751.400	220.900	100.600	98.900	11.713.700	6.398.000	3.054.900	636
14			Differenz Summe LÄNDER minus Summe_DE	1,00 %	0,00 %	0,00 %	0,00 %	0,00 %	-2,77 %	0,10 %	-2,22 %	0,00 %	0,00 %	0,00 %	-0,02 %	0,0
15			Differenz Summe KREISE minus Summe_DE	-1,00 %	0,02 %	0,02 %	0,06 %	-0,04 %	-2,43 %	-0,25 %	11,68 %	-0,09 %	-0,09 %	-0,66 %	0,7	0,7
17	1	1	Schleswig-Holstein	15	987.800	661.300	6.300	320.100	6.300	0	6.300	661.300	299.200	170.900	29	29
18	2	17	Hamburg	0	14.600	5.700	2.100	6.800	200	0	5.700	2.500	1.200	0	0	0
19	3	3	Niedersachsen	45	2.579.900	1.870.800	18.900	690.200	39.500	0	18.900	1.870.800	893.200	402.400	133	133
20	4	17	Bremen	0	8.100	1.600	0	6.400	0	0	1.600	700	0	0	0	
21	5	5	Nordrhein-Westfalen	53	1.493.300	1.064.300	12.900	415.800	28.800	0	12.900	1.064.300	584.500	250.800	24	24
22	6	6	Hessen	28	766.800	467.300	6.100	293.300	16.800	3.500	2.600	467.300	314.600	159.400	16	16

Im Hintergrund ist eine stark verkleinerte Ansicht des gesamten Berechnungsblattes zu sehen, im Vordergrund die linke obere Ecke in lesbarer Größe.

Quelle: eigene Darstellung, PIKOBAYTES GmbH

Da sich während der Projektlaufzeit herausgestellt hat, dass die erwartete Konsistenz nicht durchgehend vorhanden ist, musste die Referenzimplementierung in einem zweistufigen Prozess abgeleitet werden. Details dazu werden in den Kapiteln 3.3.2 und 3.3.3 näher beschrieben und wurden dem UBA in einem separaten Dokument dokumentiert.

3.2.2 Referenzimplementierung Oktober/2024

Fast zeitgleich mit der Finalisierung der Neu-Implementierung der Kernberechnung im Oktober 2024 wurde durch Herrn Häußermann die nächste Ausgabe der Stickstoff-Flächenbilanz in Form der Datei *N-Flächenbilanzen Kreise Berechnung 1995-2022_Update_16_10_24.xlsm* bereitgestellt. Die Vorstellung der Ergebnisse und der methodischen Anpassungen erfolgte im Dezember 2024.

Während es in der Kernberechnung nur geringfügige Abweichungen zur Referenzimplementierung gibt, ist die Datenvorverarbeitung mit dieser Ausgabe komplett umgestellt worden.

Der ursprüngliche Plan im Projekt sowohl die Kernberechnung als auch die Datenvorverarbeitung anhand der Referenzimplementierung von 2023 neu zu implementieren wurde daher verworfen. Es entspricht nicht der Zielstellung des Projektes, ein veraltetes Modell nachzubauen. Stattdessen haben alle Projektbeteiligten gemeinsam beschlossen, dass stattdessen ab sofort die neue Version als Referenzimplementierung insbesondere für die Datenvorverarbeitung genutzt werden sollte und die kleineren Anpassungen in der Kernberechnung entsprechend nachgepflegt werden. Da es eine saubere Trennlinie zwischen Kernberechnung und Datenvorverarbeitung gibt, ist es praktisch ausgeschlossen, dass ein Wechsel des Vorverarbeitungsmodells Seiteneffekte auf die Funktionalität oder die Ergebnisse der Kernberechnung hat.

Die Excel-Dateien die ab Dezember 2024 die Referenzimplementierung für die Datenvorverarbeitung darstellen wurden in mehreren Tranchen im Dezember 2024, Mitte Februar 2025 und Ende Juni 2025 zur Verfügung gestellt.

3.3 Entwicklung von Werkzeugen zur Analyse der Referenzimplementierung

3.3.1 Extraktionswerkzeuge

Aus der Referenzimplementierung müssen aus den einzelnen Tabellenblättern folgende Informationen extrahiert werden (vgl. Kapitel 3.2.1):

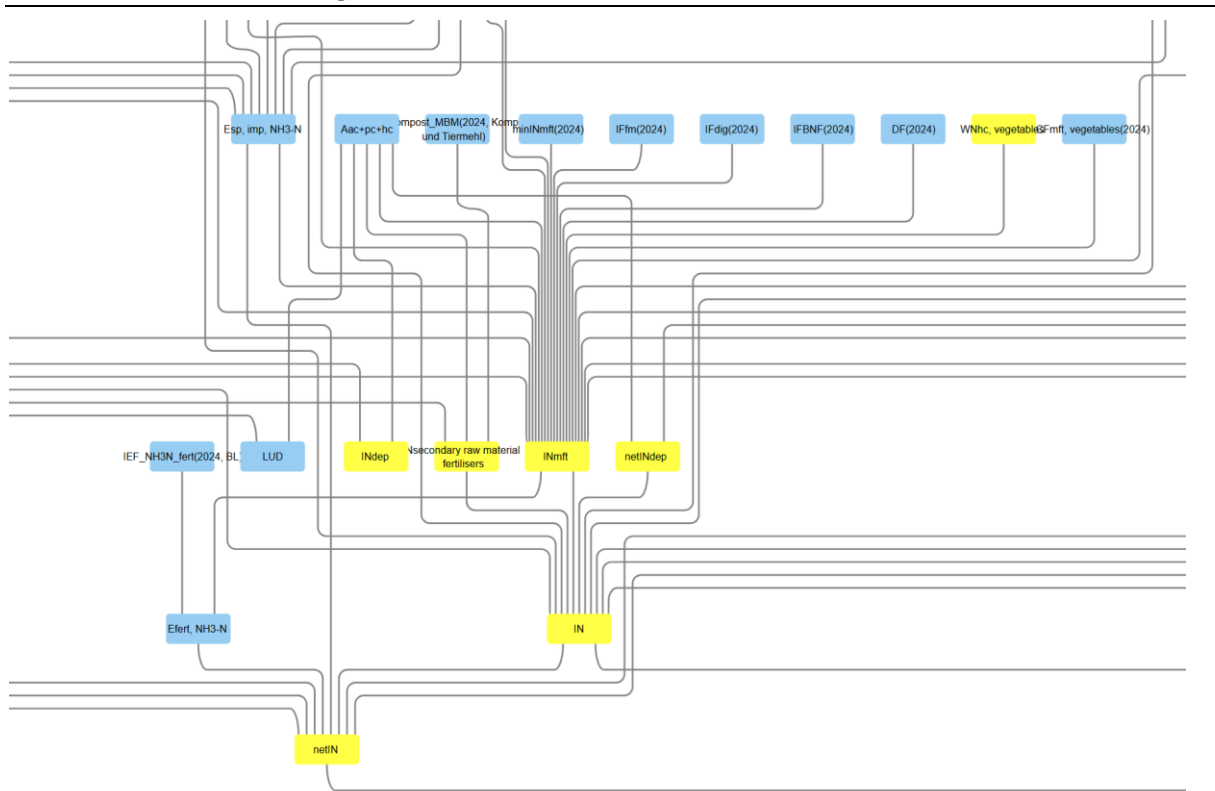
- ▶ numerische Eingangsdaten, Zwischen- und Endergebnisse in der Stickstoffbilanzierung,
- ▶ Rechenformeln in den einzelnen Zellen,
- ▶ strukturelle Verknüpfungen zwischen Spalten, Zeilen und Rechenblättern, aus denen Datenflüsse und Rechenwege abgeleitet werden können,
- ▶ numerische Konstanten und statische Parameter, die in die Berechnung einfließen.

Während die eigentlichen Berechnungsformeln nicht direkt aus Excel extrahiert werden können, sondern händisch analysiert und nachimplementiert werden mussten, konnten die numerischen

Werte automatisch extrahiert, transformiert und in der transformierten Form durch die nachimplementierte Berechnung automatisch eingelesen werden.

Als erstes wurden deshalb Python-Programme entwickelt, die alle Konstanten und statischen Parameter automatisch detektieren, auslesen und in einem definierten Format ablegen. Die Stickstoffflächenbilanzierung ist grundsätzlich so aufgebaut, dass in einem mehrstufigen Prozess die Stickstoffzufuhr und die Stickstoffabfuhr im Sektor Landwirtschaft berechnet werden. Das sind die beiden Seiten der Bilanz, aus denen sich der Saldo, also Überschuss oder Verlust, ergibt. Mehrstufig heißt, dass sich Zufuhr und Abfuhr aus mehreren Teilbeträgen zusammensetzen, die sich wiederum aus Zwischengrößen ergeben. Diese Zwischengrößen sind in der Regel auch wieder mehrstufig definiert und werden über weitere Zwischenstufen aus den Eingangsdaten abgeleitet. Abbildung 5 verdeutlicht dieses Prinzip anhand eines Auszuges.

Abbildung 5: Schematische Darstellung zur mehrstufigen Ableitung von Zwischengrößen zur N-Bilanzierung.



Quelle: eigene Darstellung, PIKOBYTES GmbH

Die Berechnung erfolgt für jede Zwischengröße pro Region (Großkreis) und pro Kalenderjahr. Die meisten der in die Berechnung einfließenden statischen Parameter liegen tabellarisch vor, in der Regel ist es dann ein Wert pro Region und Jahr. Region kann dabei heißen: Für jedes Bundesland liegt ein spezifischer Wert vor oder für jede Region (also auch für jeden Kreis) liegt ein spezifischer Wert vor. Einige statische Parameter liegen als Zeitreihe vor, einige als regionalspezifische Konstanten und einige als konstante Einzelwerte, die für alle Kreisregionen gleichermaßen gelten. Diese Konstanten und statischen Parameter sind in der Referenz-Rechenmappe auf folgende Tabellenblätter verteilt: "EFK", "RMD", "YML", "EFL", "YGL", "IMP", "SBI", "DEP", "WDI". Die Herausforderung lag darin, dass auf vielen dieser Tabellenblätter mehrere kleine Tabellen mit eben jenen Werten untergebracht waren. Diese eingebetteten

Tabellen sind aufgrund der oben beschriebenen Charakteristik der Daten in ihrer Struktur grundsätzlich ähnlich. Diese Ähnlichkeit ist jedoch nur oberflächlich, die Daten selbst sind im Detail wieder unterschiedlich angeordnet, der Datenbereich ist unterschiedlich groß, und die Tabellen haben eine unterschiedliche Anzahl an leeren Zeilen oder Spalten, unterschiedliche Zeilenbezeichnungen oder unterschiedliche Anzahlen an Kopfspalten. Erschwerend kam hinzu, dass die Kennzeichnungen der jeweiligen Kenngrößen in den Tabellen unterschiedlich codiert sind und an unterschiedlichen Positionen geführt werden.

Einige Konstanten bzw. Parameter werden nicht an zentraler Stelle definiert und befinden sich im Tabellenkopf der Berechnungsblätter. Auch diese Werte wurden extrahiert.

Letztendlich sind für die Extraktionsaufgaben zwei Python-Skripte entstanden:

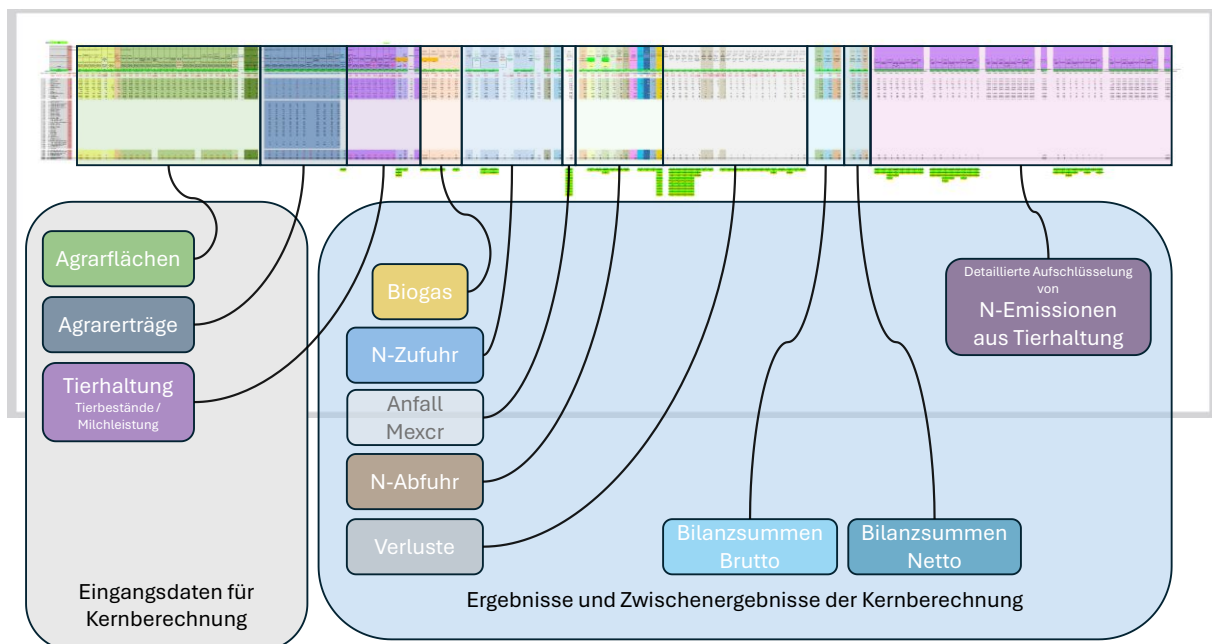
- ▶ *tables_in_sheets.py* – mit der Logik zur Detektion und Extraktion der Konstanten und Tabellen sowie
- ▶ *extract_config_tables.py* – ein kleines Programm welches die tabellenblattspezifische Konfiguration (z. B. Suchbereich und welche Art von Tabellen gesucht wird) enthält und die Ausführung steuert.

Die damit verbundenen Arbeiten wurden zum Großteil im Zeitraum Mai 2023 bis November 2023 durchgeführt. Die grundlegende Implementierung war Mitte Juni fertiggestellt. Da projektbegleitend die Kernberechnungen und Rechenmappen fortgeschrieben und überarbeitet wurden, wurden ab September 2023 neue Funktionen für anfangs nicht unterstützte Tabellenblätter ergänzt und im Zuge von Validierungen einzelne Fehler behoben.

Ein weiteres Werkzeug zur Datenextraktion aus der Excel-Rechenmappe ist das Script *ber_sheet_to_csv.py*. Mit dessen Hilfe können aus den Berechnungsblättern sowohl die Eingangsdaten der Berechnung als auch die Ergebnisse der Berechnung extrahiert werden. Erstere werden als Eingangsdaten für die Berechnung mit der Python-Implementierung

verwendet, letztere zur Qualitätskontrolle für den Vergleich mit den Ergebnissen der Python-Implementierung.

Abbildung 6: Überblick der Bereiche in den Berechnungsblättern



Hier ist ein stark verkleinertes Abbild des Tabellenblatts *2019 ber* zu sehen, welches als Referenz für die Nachimplementierung dient – vgl. auch Abbildung 4. Gleichfarbig überlagert sind Zwischengrößen / Spalten, die zur selben Kategorie gehören. (Die Tabelle erscheint hier deutlich kürzer als sie tatsächlich ist, weil ein Großteil der Kreisregionen in den Zeilen ausgeblendet ist)

Quelle: eigene Darstellung, PIKOBAYTES GmbH

Wesentliches Ergebnis dieser Projektphase sind Werkzeuge und Mechanismen zur Strukturierung und Ablage von Konstanten und statischen Parametern, sodass sie später während der Berechnung direkt geladen werden können (vgl. auch Kapitel 3.4.4.2.1).

3.3.2 Werkzeuge zur Konsistenzprüfung

Im Zuge der ersten Implementierung der Kernberechnung anhand des Referenz-Tabellenblattes "2019 ber" wurde zur Qualitätssicherung ein relativ einfacher zahlenmäßiger Vergleich zwischen den Ergebnissen der Referenzimplementierung und den Ergebnissen der Python-Implementierung umgesetzt. Dieser erschien zunächst ausreichend.

Die so erstellte Neu-Implementierung der Kernberechnung lieferte für das Jahr 2019 mit der Referenzimplementierung weitestgehend übereinstimmende Ergebnisse. Bei der Anwendung dieses Referenzrechenwegs auf die übrigen Jahre traten jedoch größere quantitative Abweichungen zwischen den Ergebnissen der Referenz und den Ergebnissen der Python-Implementierung auf.

In der Überprüfung wurde festgestellt, dass die Formeln in den Berechnungsblättern nicht wie erwartet durchgängig konsistent waren. Stichprobenartig konnte gezeigt werden, dass sich die Rechenwege zwischen den einzelnen Jahren teils stark voneinander unterschieden. Da sich die Rechenwege zwischen den Jahren nicht grundlegend ändern sollten und von vornherein nicht klar war, welcher der abweichenden Rechenwege „der richtige“ ist, war eine Detailprüfung der Berechnungsvorschriften und eine systematische Überprüfung der Formeln in den

Rechenblättern erforderlich. Aufgrund der schier unendlichen Anzahl an Formeln⁵ war eine manuelle Prüfung hier nicht mehr möglich.

Für die Überprüfung der Formeln wurden zusätzliche Python-Werkzeuge entwickelt, die Excel-Tabellenblätter hinsichtlich der Formeln und zahlenmäßigen Werte vergleichen und geeignete Auswertungen bereitstellen, um Abweichungen leicht zu erkennen und mit den Entwicklern des bisherigen Modells zu diskutieren und zu bereinigen. Diese Werkzeuge werden nachfolgend beschrieben.

3.3.2.1 Werkzeug „compare_pflast_files“

Dieses Werkzeug stellt die Möglichkeit bereit, wahlweise

- ▶ Tabellenblätter innerhalb einer Excel-Datei (Blattvergleich) oder
- ▶ gleichnamige Tabellenblätter zweier Excel-Dateien miteinander (Änderungsprotokoll) zu vergleichen.

Als Ergebnis wird ein Protokoll ausgegeben, in dem die Unterschiede detailliert dargestellt werden. Ähnliche Unterschiede werden zusammengefasst, um besonders bei vielen betroffenen Zellen die Lesbarkeit und Übersichtlichkeit zu erhalten. Die Ausgabe des Protokolls erfolgt in den gleichen Formaten wie die Analysedokumentation (Markdown mit eingebetteten Latexausdrücken) sowie als PDF (durch Umwandlung des Markdownprotokolls mit Hilfe von Pandoc) und kann damit sowohl maschinenlesbar weiterverarbeitet als auch visuell interpretiert bzw. gelesen werden.

Folgende Dokumente wurden mit diesem Werkzeug erstellt, die jeweils zwei Versionen der Referenzimplementierung gegenüberstellen (links die Ausgangsdatei, rechts die geänderte Version). Der Dateiname des Protokolls enthält jeweils nur den Dateinamen der linken Datei. Im Änderungsprotokoll werden beide Dateinamen genannt:

- ▶ Änderungsprotokoll (Daten) für N-Flächenbilanzen Kreise Berechnung 1995-2021_Update_17_2_23_DüngEval.xlsm_-.pdf
 - Linke Datei: originale Referenzimplementierung von 2023
 - Rechte Datei: originale Referenzimplementierung von 2023 neu berechnet
- ▶ Änderungsprotokoll (Daten) für N-Flächenbilanzen Kreise Berechnung 1995-2021_Update_17_2_23_DüngEval_recomputed.xlsm_-.pdf
 - Linke Datei: originale Referenzimplementierung von 2023 neu berechnet
 - Rechte Datei: fehlerbereinigten Referenzimplementierung von 2023
- ▶ Änderungsprotokoll (Formeln) für N-Flächenbilanzen Kreise Berechnung 1995-2021_Update_17_2_23_DüngEval.xlsm_-.pdf
 - Linke Datei: originale Referenzimplementierung von 2023
 - Rechte Datei: originale Referenzimplementierung von 2023 neu berechnet

⁵ Die 29 Berechnungsblätter enthalten je 145 Werte-Spalten, von denen 102 Spalten berechnet werden sowie 418 Zeilen mit Kreisregionen. Das wären also $102 \times 418 \times 29 = 1236444$ teilweise sehr umfangreiche Formeln, was eine manuelle Prüfung ausschließt.

- ▶ Änderungsprotokoll (Formeln) für N-Flächenbilanzen Kreise Berechnung 1995-2021_Update_17_2_23_DüngEval_recomputed.xlsm_-.pdf
 - Linke Datei: originale Referenzimplementierung von 2023 neu berechnet
 - Rechte Datei: fehlerbereinigten Referenzimplementierung von 2023

Das Vergleichswerkzeug bietet folgende Einstellungsmöglichkeiten:

Tabelle 3: Einstellungsmöglichkeiten des Vergleichswerkzeuges

Modus	Option	Blattvergleich	Änderungsprotokoll
Dateivergleichsmodus	Anzahl Dateien	1	2
Datenmodus	Formeln	x	x
Datenmodus	Daten (Werte)	x	x
Reportmodus	nach Blatt und Spalte	x	x
Reportmodus	nach Spalte und Blatt	x	x
Blattvergleichsmodus	gleichnamig	-	x
Blattvergleichsmodus	Vorgänger / Nachfolger	x	-
Blattvergleichsmodus	eins gegen die anderen	x	-

Report **nach Blatt und Spalte** bedeutet, die Ergebnisse werden jeweils für ein Blatt (zum Beispiel 2019 ber) zusammengefasst. Je Blatt erfolgt dann die Auflistung der Unterschiede für die einzelnen Spalten. Das bietet sich besonders für das Änderungsprotokoll an.

Report **nach Spalte und Blatt** bedeutet, die Ergebnisse werden jeweils für eine Spalte (zum Beispiel IN) zusammengefasst. Je Spalte erfolgt dann die Auflistung der Unterschiede für die einzelnen Blätter / Jahre.

gleichnamig - Gegenüberstellung anhand Blattname bedeutet, es werden jeweils gleichnamige Blätter miteinander verglichen, zum Beispiel 2019 ber aus Datei A und aus Datei B.

Vorgänger / Nachfolger - Gegenüberstellung von aufeinanderfolgenden Blättern bedeutet, es werden jeweils zwei aufeinanderfolgende Blätter miteinander verglichen, zum Beispiel 1995 ber vs. 1996 ber, 1996 ber vs. 1997 ber, ..., 2019 ber vs. 2020 ber.

eins gegen die anderen - Gegenüberstellung eines Blattes gegen jedes andere ausgewählte Blatt bedeutet ein Blatt wird als Referenz ausgewählt und alle anderen Blätter werden mit diesem Blatt verglichen, zum Beispiel 2019 ber vs. 1995 ber, 2019 ber vs. 1996 ber, ..., 2019 ber vs. 2020 ber

Die Benutzeroberfläche ist in Abbildung 7 dargestellt.

Abbildung 7: Benutzeroberfläche *compare_pflast_files*

Quelle: eigene Darstellung, PIKobytes GmbH

3.3.2.2 Werkzeug „formula_analysis“

Die Aufgabe dieses Werkzeugs ist es, zwei Versionen der N-Bilanzierungsrechenmappe hinsichtlich ihrer Formeln zu vergleichen. Es wurde entwickelt, um vorgenommene Modifikationen und Nachkorrekturen an den Referenzdateien (1) nachverfolgen und validieren

zu können, und (2) um Formelabweichungen aufzuspüren. Insbesondere diente es dem Nachweis, dass die Korrektur der Excel-Datei (Kapitel 3.3.3) erfolgreich war.

Dieses Werkzeug ermittelt für jede Spalte bzw. Zwischengröße der Berechnungsblätter über alle Jahre und Kreisregionen hinweg alle *Varianten* der in der jeweiligen Spalte verwendeten *Berechnungsformeln* und erstellt entsprechende Formelübersichten.

3.3.2.2.1 Formelvarianten

Formel-Varianten sind abweichende Rechenvorschriften in den Referenzrechenmappen. Die Formeln zur Berechnung einer Zwischengröße müssen in Excel innerhalb einer Spalte in jeder Zeile wiederholt werden; die Zellbezüge in den Formeln ändern sich von Zeile zu Zeile. Diese Unterschiede sind bei der Nutzung von Tabellenkalkulationsprogrammen prinzipbedingt und so erwartet. Darüber hinaus gibt es oft unterschiedliche Berechnungsformeln für verschiedene Regionalebene, also verschiedene Formeln für die Bundesländer und die Kreise. Diese Unterschiede sind in einigen Fällen auf Eigenschaften in den Quelldaten zurückzuführen. Für manche Zwischengrößen ändern sich deshalb ab einem bestimmten Kalenderjahr die Rechenvorschriften.

Durch die Arbeitsweise in Excel entstehen jedoch auch sehr leicht Kopierfehler oder unbeabsichtigte Unterschiede in den Berechnungsformeln. Diese Abweichungen sind als strukturelle Fehler zu werten, da sie vom beabsichtigten Rechenweg abweichen. In diesen Fällen muss festgestellt werden, wo eine ungewollte Abweichung besteht und welche der Formelvarianten die richtige ist.

Während der Analyse der Referenzimplementierung sind auch systematische Fehler in den Formeln entdeckt worden, beispielsweise ein doppelt vorhandener Faktor. Die Behebung dieser Fehler ergibt letztendlich eine neue Formelvariante, die mit den Analysewerkzeugen nachverfolgt und zu Prüf- und Kontrollzwecken kenntlich und nachverfolgbar gemacht werden muss.

3.3.2.2.2 Formel-Muster

Für die Formelvergleiche ist eine Umwandlung der ursprünglichen Formeln in ein Formel-Muster erforderlich, das einen Vergleich ohne die erwartbaren kleinen zeilenbezogenen Unterschiede in den Formeln ermöglicht. Dafür werden sämtliche Zeilenbezüge gegen geeignete Platzhalter (z.B. {row}, {row_before}) ausgetauscht. Ein so codiertes Formel-Muster sieht so aus:

```
=IF(D{row}>0,M{row}-
(N{row}+W{row}+X{row}+AA{row}+AC{row}+AD{row}), "")
```

In dieser Form lassen sich die Formelvarianten zellenübergreifend leicht vergleichen.

Das Vergleichswerkzeug identifiziert distinkte Formelvarianten aus beiden verglichenen Dateien. Für die visuelle Analyse weist es jeder Formelvariante eine eigene Farbe zu, sodass schnell erkannt werden kann, welche Formelvariante wo verwendet wird.

Die erzeugten visuellen Auswertungen machen die an den Referenzrechenblättern angebrachten Korrekturen sowie die auf Grundlage der Arbeiten in Kapitel 3.3.2.1 angebrachten Änderungen zur Vereinheitlichung der Rechenwege nachvollziehbar.

3.3.2.2.3 Formelvergleichsübersicht

In der Formelvergleichsübersicht wird die Verteilung der Formelvarianten einer berechneten Größe über die Jahre und Kreisregionen für die beiden verglichenen Dateien gegenübergestellt (Abbildung 8). Eine solche Übersicht wird für jede Spalte / jede berechnete Größe erzeugt.

Rechenwege visuell unterscheidbar machen. Die Zuordnung zu den Formelvarianten ist aus der Legende unterhalb der Tabelle ersichtlich. So können für einzelne Jahre oder Kreisregionen abweichende Rechenvorschriften schnell identifiziert werden. Weiterhin ist erkennbar, dass zu bestimmten Jahren ein Wechsel der Rechenvorschriften stattgefunden hat. In diesem Beispiel sind sehr gut einige Anomalien in der Originaldatei für das Jahr 2012 erkennbar, wo drei Bundesländer und einige Kreise in Mecklenburg-Vorpommern deutlich von den anderen Kreisregionen abweichende Formeln aufweisen. Es ist auch deutlich sichtbar, dass im Jahr 2010 auf Kreisebene ein Wechsel der Berechnungsvorschrift stattfand, der auch nach der Korrektur erhalten geblieben ist. Wohingegen auf Bundeslandebene die original vorhandenen Wechsel in den Berechnungsvorschriften nach der Korrektur nicht mehr vorhanden sind.

Quelle: eigene Darstellung, PIKOBYTES GmbH

3.3.2.2.4 Formeltabellen – Detailsicht

Neben der Formelvergleichsübersicht gibt das Werkzeug auch eine Detailübersicht aus. In dieser Darstellung sind die Formelvarianten nicht numerisch codiert; stattdessen steht hier die vollständige Formel in ausgeschriebener Form. Unterschiedliche Formelvarianten sind auch hier wieder farblich codiert. Die Farbcodierung stimmt mit der Formelvergleichsübersicht (Abbildung 8) exakt überein, sodass beide Analyseergebnisse kombiniert genutzt werden können.

Die Detailsichten lassen sich aufgrund der Länge der Formeln nicht kompakt darstellen. Sie werden deshalb als HTML-Dateien für die Ansicht im Browser generiert (Abbildung 9). Diese Detailsicht wird für jede Zwischengröße und jeweils für die Originaldatei und die korrigierte Datei erstellt.

Abbildung 9: Beispiel eine Formeltabelle für eine Spalte

	Aac, other arable land:AE-1995	Aac, other arable land:AE-1996	Aac, other arable land:AE-1997	Aac, other arable land:AE-1998	Aac, other arable land:AE-1999	Aac, other arable land:AE-2000	Aac, other arable land:AE-2001	Aac, other arable land:AE-2002	Aac, other arable land:AE-2003	Aac, other arable land:AE-2004	Aac, other arable land:AE-2005	Aac, other arable land:AE-2006	Aac, other arable land:AE-2007	Aac, other arable land:AE-2008	Aac, other arable land:AE-2009	Aac, other arable land:AE-2010	Aac, other arable land:AE-2011	Aac, other arable land:AE-2012	Aac, other arable land:AE-2013	A
13 (DG Deutschland)	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "	=IFD13-0, M13 - N13 + Y13 - AA13 + AC13 + AD13, "
17 (01 Schleswig Holstein)	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "	=IFD17-0, M17 - N17 + Y17 - AA17 + AC17 + AD17, "
18 (02 Hamburg)	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "	=IFD18-0, M18 - N18 + Y18 - AA18 + AC18 + AD18, "
19 (03 Niedersachsen)	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "	=IFD19-0, M19 - N19 + Y19 - AA19 + AC19 + AD19, "
20 (04 Bremen)	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "	=IFD20-0, M20 - N20 + Y20 - AA20 + AC20 + AD20, "
21 (05 Nordrhein Westfalen)	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "	=IFD21-0, M21 - N21 + Y21 - AA21 + AC21 + AD21, "
	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "	=IFD22-0, M22 - N22 + Y22 - AA22 + AC22 + AD22, "

Die Abbildung zeigt einen Ausschnitt aus der detaillierten Formelübersicht für Aac,arable land (sonstige landwirtschaftliche Anbaufläche). Es sind die in der originalen Referenzimplementierung verwendeten Formelvarianten über alle Jahre und Kreisregionen in einer Tabelle dargestellt. Jeder Formelvariante ist eine eigene Farbe zugeordnet, so dass schnell ersichtlich ist, wo verschiedene Rechenwege zum Einsatz kommen.

Quelle: eigene Darstellung, PIKOBYTES GmbH

3.3.3 Werkzeuge zur Nachführung und Fehlerbehebung in der Referenzimplementierung

Mit den in Kapitel 3.3.2 beschriebenen Werkzeugen zur Konsistenzprüfung konnten Abweichungen und Inkonsistenzen in den Rechenwegen nachgewiesen werden. In Rücksprache mit der Arbeitsgruppe Bach (den Autoren der Referenzimplementierung) konnte geklärt

werden, welche Berechnung beabsichtigt ab einem bestimmten Zeitpunkt auf eine andere Weise erfolgt und welche Formeln korrigiert werden müssen, um Fehler zu beseitigen.

In Anbetracht der großen Zahl an erforderlichen Anpassungen und Änderungen wurde eine manuelle Anpassung der Excel-Referenzdatei wegen der großen Fehleranfälligkeit nicht in Betracht gezogen. Stattdessen wurden die folgenden zwei Python-Module entwickelt, die die Excel-Datei automatisch und reproduzierbar korrigieren.

Im Ergebnis lag dann eine nachkorrigierte Referenzimplementierung vor. In einem gemeinsamen Arbeitstreffen mit dem UBA und der AG Bach am 21. August 2024 wurden die Unterschiede zwischen der korrigierten und der originalen Referenzimplementierung vorgestellt und die sich daraus ergebenden Änderungen in den Berechnungsergebnissen bewertet. Ab diesem Zeitpunkt – August 2024 – wurde die korrigierte Version als neue Arbeitsgrundlage⁶ für die Re-Implementierung genutzt.

3.3.3.1 Werkzeug `create_patches`

Dieses Werkzeug enthält manuell erstellte Template-Definitionen mit deren Hilfe konkrete zellgenaue Änderungskonfigurationen (Patches) erstellt werden. Diese können dann auf eine bestehende Referenz-Rechenmappe angewendet werden, um systematische Anpassungen an den Berechnungsvorschriften vorzunehmen. Im konkreten Fall wurde dieses Werkzeug genutzt, um Änderungsvorschriften zu erstellen, die auf die Datei „N-Flächenbilanzen Kreise Berechnung 1995-2021_Update_17_2_23_DüngEval_recomputed.xlsx“⁷ angewendet werden sollen, um die aus der Analyse bekannten Fehler zu korrigieren.

3.3.3.2 Werkzeug `patch_excel_file`

Dieses Werkzeug wendet die von `create_patches` erstellten zellgenauen Änderungsanweisungen auf die zu korrigierende Excel-Datei an. Dabei wird die Excel-Datei nicht verändert, sondern eine neue Datei mit den geänderten Formeln erstellt.

3.3.4 Werkzeuge zur Analyse und Veranschaulichung des Rechenweges

Das Werkzeug `colum_dependency` erstellt einen Graphen, der die Zusammenhänge zwischen den Eingabedaten und den berechneten (Zwischen-)Größen der N-Flächenbilanzierung darstellt und als Graphml⁸-Datei ausgegeben wird. Das ermöglicht es, die Zusammenhänge zwischen den einzelnen Größen zu veranschaulichen und nachzuvollziehen, ohne die Rechenvorschriften zu analysieren.

3.4 Umsetzung des Rechenkerns zur Stickstoffbilanzierung

3.4.1 Überführung der Rechenvorschriften aus der Referenzimplementierung in Python-Code

Wie zuvor bereits benannt, können die Excel-Formeln nicht direkt nach Python portiert und auch nicht automatisch konvertiert werden. Um eine sinnvolle Berechnung in Python zu implementieren, ist es notwendig den Rechenweg zu verstehen und in geeigneter Weise für die

⁶ Der entsprechende Dateiname lautet "N-FlächenbilanzenKreiseBerechnung1995-2021_Update_17_2_23_DüngEval_korrigiert.xlsx"

⁷ Diese basiert auf der ursprünglichen Referenzdatei „N-Flächenbilanzen Kreise Berechnung 1995-2021_Update_17_2_23_DüngEval.xlsx“ vom 11. Juli 2023.

⁸ <http://graphml.graphdrawing.org/>

Python-Umgebung neu aufzubauen. Dafür war es notwendig, die Formeln auf den Berechnungsblättern ausführlich manuell zu analysieren und dokumentieren.

Diese Dokumentation erfolgte in Markdown-Dateien, aus denen mit Hilfe von Pandoc PDF-Dateien generiert werden.

3.4.2 Technologiestack

Die Neuimplementierung der N-Flächenbilanzierung erfolgt mit Python, insbesondere unter Verwendung von Pandas DataFrames⁹.

Zur Ausführung wird mindestens Python 3.13 vorausgesetzt. Als wesentliche Bibliothek wird Pandas in der aktuellen Version 2.3.3 verwendet. Daneben werden zahlreiche weitere Python-Packages eingebunden, die in der pyproject.toml-Datei deklariert sind. Für das Package-Management wird im Moment Poetry¹⁰ eingesetzt.

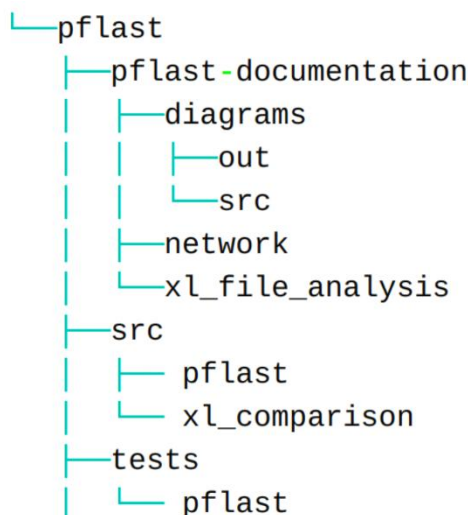
Pandas wird in absehbarer Zeit in einer neuen Hauptversion bereitstehen. Eine Migration auf Pandas Version 3 sollte vollzogen werden, sobald sich diese neue Version als stabil erweist. Hierfür werden voraussichtlich nur geringfügige Anpassungen an den bereitgestellten Modulen und Anwendungen erforderlich sein.

Zum Bearbeiten des Projektes können IDEs wie Visual Studio Code oder PyCharm (in der jeweils aktuellen Version) oder erweiterte Texteditoren wie Vim oder Emacs verwendet werden.

3.4.3 Struktur des Softwareprojektes

Nachfolgend wird die grobe Struktur des Softwareprojektes beschrieben. Diese soll lediglich einen Überblick zum Softwareprojekt geben. Die detaillierte Entwicklungsdokumentation ist im Softwareprojekt selbst enthalten und sollte auch dort weiter gepflegt werden.

Abbildung 10: Verzeichnisstruktur des Softwareprojektes



Zu sehen ist die Gliederung der oberen Verzeichnisebenen des Softwareprojektes.

Quelle: eigene Darstellung, PIKobytes GmbH

Das Projekt ist wie folgt aufgebaut:

⁹ <https://pandas.pydata.org/>

¹⁰ <https://python-poetry.org/>

► *pflast_documentation*

- enthält die technische Dokumentation der Referenzimplementierung (*xl_file_analysis*), das Fließschema (*network*) der Kernberechnung sowie Diagramme (*diagrams*), die zur Entwicklungsdokumentation des Softwareprojekts selbst gehören
- die Entwicklungsdokumentation ist in Form von README.md-Dateien (und Kommentaren im Code selbst) über das gesamte Softwareprojekt verteilt

► *src*

- enthält den Python-Quellcode
- *pflast* ist das Haupt-Package der N-Flächenbilanz-Python-Implementierung
- *xl_comparison* ist eine Pikobytes-Bibliothek zum Auswerten von Excel-Dateien. Diese wird zusätzlich zur Verfügung gestellt, da einige Werkzeuge für die Analyse und Verarbeitung der Excel-Dateien im aktuellen Entwicklungsstadium wahrscheinlich noch gebraucht werden.

► *tests/pflast*

- enthält Unit-Tests für den Code in *src/pflast*

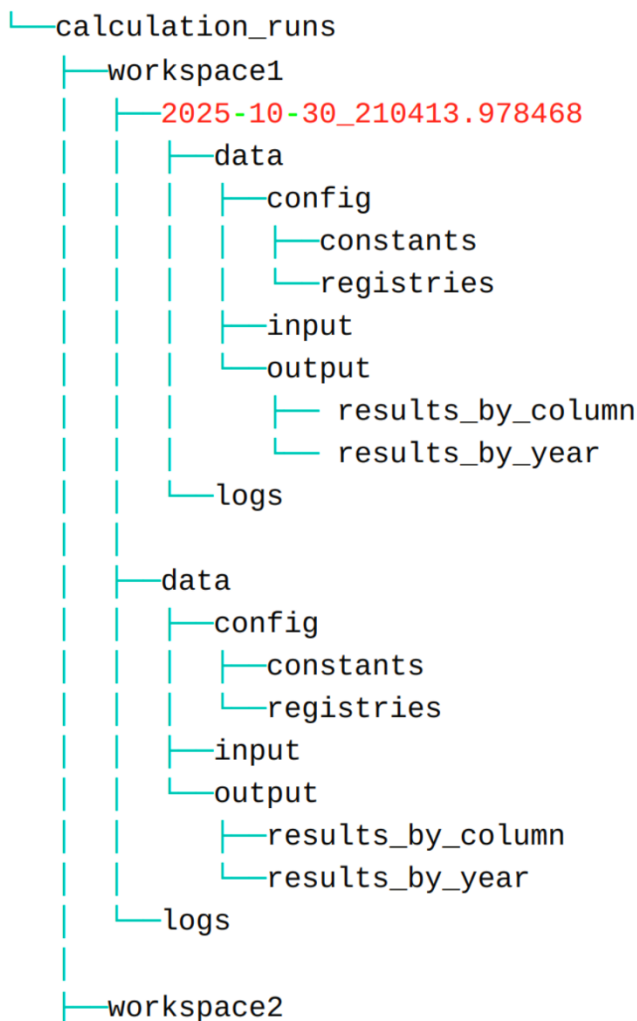
3.4.4 Workspace

In Anlehnung an das zuvor beschriebene Workspace-Konzept für die Pflege und Versionierung des Bilanzierungsmodells (Kapitel 2.6 und Abbildung 1) wird nachfolgend beschrieben, wie dieses Konzept für die Neuimplementierung umgesetzt wurde.

3.4.4.1 Workspace-Struktur

Für den Betrieb des Modells wird die folgende festgelegte Ordnerstruktur vorausgesetzt. Es gibt ein festgelegtes Verzeichnis, welches alle Workspaces beherbergt: *calculation_runs*. Unterhalb dieses Verzeichnisses sieht die Struktur wie in Abbildung 11 dargestellt aus.

Abbildung 11: Workspace-Struktur



Beispielhafte Darstellung der Ordnerstruktur unter dem Wurzelverzeichnis *calculation_runs*. Zu sehen ist ein Workspace namens *workspace1* mit den standardmäßig vorhandenen Verzeichnissen *data* und *logs* sowie – rot hervorgehoben – einem nach seinem Startzeitpunkt benannten Verzeichnis für einen Rechendurchlauf. Ein zweiter Workspace mit Namen *workspace2* ist angedeutet.

Quelle: eigene Darstellung, PIKobytes GmbH

Jedes Workspace-Verzeichnis enthält ein *data*-Verzeichnis und ein *logs*-Verzeichnis. Wahlweise wird diese Struktur für jeden Lauf in ein eigenes Unterverzeichnis kopiert, welches als Namen den Zeitstempel vom Start des jeweiligen Laufs erhält. Werden keine eigenen Unterverzeichnisse genutzt, werden die Ergebnisse des letzten Laufs im Workspace-Verzeichnis überschrieben.

3.4.4.1.1 logs-Verzeichnis

Das *logs*-Verzeichnis enthält die Log-Dateien des jeweiligen Laufes. Wenn keine laufspezifischen Unterverzeichnisse verwendet werden, können die Log-Dateien (je nach Logging-Konfiguration) entweder nur die Log-Einträge des letzten Laufs oder die Log-Einträge mehrerer Läufe enthalten.

3.4.4.1.2 data-Verzeichnis

Das *data*-Verzeichnis enthält vor dem Berechnungslauf

- ▶ die Eingangsdaten im Unterverzeichnis *input* sowie
- ▶ die Konstanten und statischen Parameter im *config*-Unterverzeichnis.

Die Ergebnisse der Berechnung werden in die *output*-Unterverzeichnisse geschrieben.

3.4.4.2 Eingangsdaten

Die Eingangsdaten werden in Form von CSV-Dateien mit der folgenden Struktur erwartet:

Spalten: Region-SNR,NUTS-Code,SNR,Zuordnung zu:,Reg_Einheit,Anzahl,Aac+pc+hc,"Aac, arable land",Apc,"Aac, grassland","Aac, extensive grassland","Apc, vineyards","Aac, cereals","Aac, winter wheat","Aac, rye","Aac, winter barley","Aac, spring barley","Aac, oats","Aac, triticale","Aac, maize","Aac, potatoes","Aac, sugar beet","Aac, root crops","Aac, silage maize","Aac, fodder production","Aac, rape seed","Aac, oilseed","Aac, legumes","Ahc, vegetables","Ahc, strawberry","Yac, winter wheat","Yac, rye","Yac, winter barley","Yac, spring barley","Yac, oats","Yac, triticale","Yac, maize","Yac, potatoes","Yac, sugar beet","Yac, rape seed","Yac, silage maize",nLU,nctl,ndc,npig,nfp,nso,nshe,nlh

Es gibt eine Datei für jedes Kalenderjahr, für das eine Berechnung durchgeführt werden soll. Die Dateinamen müssen folgender Namenskonvention entsprechen:

```
{year}-n_balance_calculation_input.csv
```

Also zum Beispiel 1995-n_balance_calculation_input.csv

3.4.4.2.1 Konstanten und statische Parameter

Sämtliche Konstanten und statische Parameter (siehe auch Kapitel 3.3.1), die in die Berechnung einfließen, befinden sich im *config*-Verzeichnis. Das *registries*-Verzeichnis enthält mindestens eine als Registry bezeichnete JSON-Datei, in der alle Konstanten und Tabellen aufgelistet sind.

Einzelwerte stehen direkt in dieser Registry, zum Beispiel:

```
"EFindirekt, leach, N2O-N": {
    "value": 0.0075
},
```

Während der Berechnung werden die Werte einfach anhand des Namens der jeweiligen Konstanten (dem Schlüsselwert in der Registry) nachgeschlagen.

Tabellarisch aufgebaute Konstanten sind in Form von CSV-Dateien abgelegt, die sich im *constants*-Verzeichnis befinden. Die Struktur ist hier individuell und wird in der Registry angegeben.

Hier ein Beispiel für einen Tabellen-Eintrag in der Registry:

```
"EFsp, N2O-N": {
    "file_name": "calculation_runs/1a/2025-10-30_173206.510606/data/config/constants/EFL-EFsp, N2O-N.csv",
    "columns_key": "Jahr",
    "rows_key": "Region"
},
```

Das Laden der Werte während der Berechnung erfolgt anhand des Schlüsselwerts in der Registry für die jeweilige Konstante und je einem Wert für Spalte und Zeile sofern beide vorhanden sind. Eine genauere Beschreibung dieses Mechanismus ist Bestandteil der Projektdokumentation.

3.4.5 Implementierung der Berechnung

Die Rechenvorschriften werden in der Python-Implementierung einmalig definiert und dann für die einzelnen Jahre und Variablen immer wieder verwendet. Dadurch können die Berechnungsvorschriften im Vergleich zur alten Excel-basierten Implementierung¹¹ übersichtlich gepflegt und bei Bedarf modifiziert und erweitert werden. Die neue Implementierung enthält einige Basisklassen, die es ermöglichen, die Rechenvorschriften für ganze Spalten wie einfache mathematische Ausdrücke zu schreiben. Die Details dazu befinden sich in der Projektdokumentation. Nachfolgend zwei Beispiele:

Die Excel-Formel

```
=WENN(A13<>B13;"";CC13-BL13-SUMME(CR13:CS13)-CU13)
```

wird so ausgedrückt:

```
Cols.M_EXCR - (Cols.GROSS_IN_FM + Cols.E_HO_NH3N + Cols.E_ST_NH3N
+ Cols.E_MM_OTHN)
```

Die Excel-Formel

```
=IF(D17>0;
AV17-(
  AW17*@INDEX(meanLUK_ctl;MATCH($A17;meanLUK_ctl_R);$BD$484)
+AY17*@INDEX(meanLUK_pig;MATCH($A17;meanLUK_pig_R;0);$BD$486)
+BB17*@INDEX(meanLUK_she;MATCH($A17;meanLUK_she_R;0);$BD$488)
+BC17*LUK_pou
); "");
```

wird so ausgedrückt:

```
Cols.N_LU
- Cols.N_CTL * BerLookups.LUK_MEAN_CTL.get(year)
- Cols.N_PIG * BerLookups.LUK_MEAN_PIG.get(year)
- Cols.N_SHE * BerLookups.LUK_MEAN_SHE.get(year)
- Cols.N_NLH * BerLookups.LUK_POU.get(year),
```

3.5 Entwicklung von Werkzeugen zur Validierung des Rechenkerns

In Anlehnung an die Entwicklung von Werkzeugen zur Analyse der Referenzimplementierung (vgl. Kapitel 3.3) sind ähnliche Werkzeuge zur Validierung des neu entwickelten Rechenkerns erforderlich. Diese erfolgt ausschließlich quantitativ, da ein automatisierter Vergleich von Formeln und Rechenwegen zwischen der Python-Implementierung und der Verformelung in den Excel-Rechenmappen nicht direkt möglich ist.

3.5.1 Werkzeug result_comparison

Das Werkzeug *result_comparison* vergleicht jeweils zwei Ergebnisse der N-Flächenbilanz-Berechnung nach Häußermann et al. miteinander.

Eingangsgrößen für das Werkzeug sind

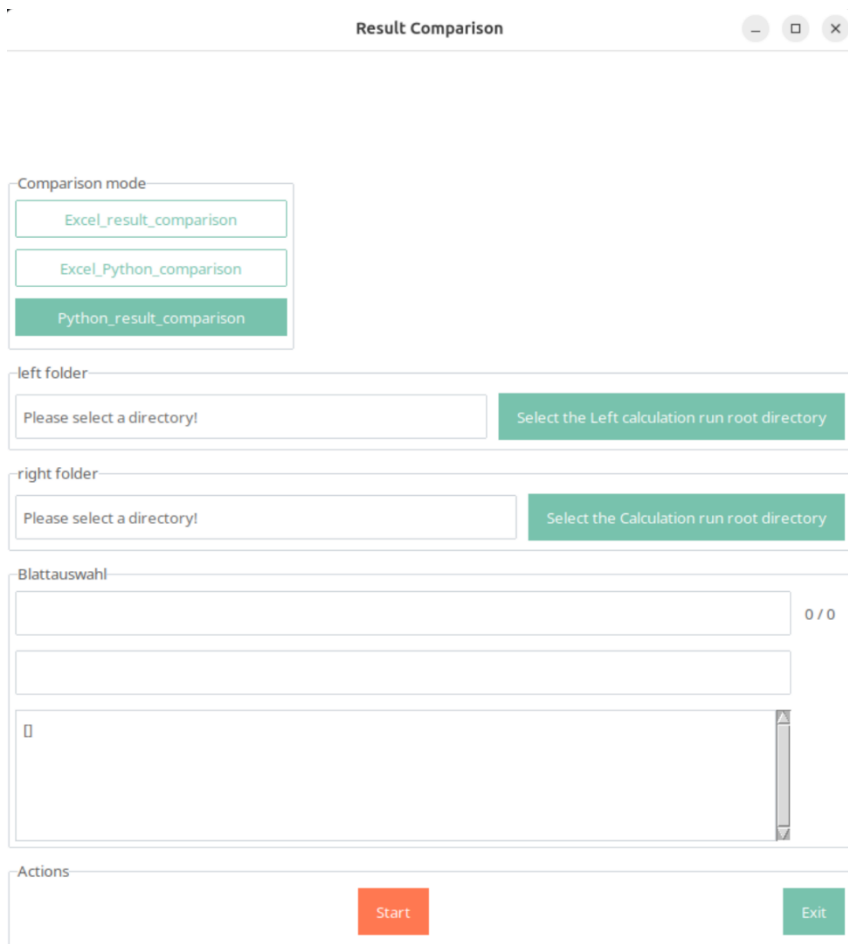
- ▶ Berechnungsblätter einer Excel-Datei zur N-Bilanzierung
- ▶ Zielverzeichnis der Berechnung der Python-Implementierung

¹¹ In der Excel-Datei kann eine einzelne Formel in einem Berechnungsblatt unter Umständen bis zu 418-mal in leicht abgewandelter Form auftreten – eine Formel für jede Region. Multipliziert mit der Zahl der Tabellenblätter sind das bis zu 30 x 418 = 12540 kopierte Formeln, die aufgrund von Kopierfehlern schwer zu erkennende Fehler enthalten können.

Es sind damit folgende Vergleiche möglich:

- ▶ Vergleich der Ergebnisse zweier Versionen der N-Bilanzierungsrechenmappe
- ▶ Vergleich der Ergebnisse einer N-Bilanzierungsrechenmappe mit dem Ergebnis eines Laufs der Python-Implementierung
- ▶ Vergleich der Ergebnisse zweier Läufe der Python-Implementierung

Abbildung 12 : Benutzeroberfläche *result_comparison*



Das Werkzeug *result_comparison_ui* stellt eine einfache Bedienoberfläche für das Werkzeug *result_comparison* bereit, über welche die Einstellungen für den Ergebnisvergleich vorgenommen werden können

Quelle: eigene Darstellung, PIKobytes GmbH

Die Ausgabe des Ergebnisvergleichs erfolgt aufgliedert nach Spalten in die folgenden Dateien:

Tabelle 4: Übersicht über Dateiausgabe des Ergebnisvergleichs

Dateiname	Inhalt
`\${Spalte}_original.csv	eine CSV-Datei mit den Werten der jeweiligen Spalte aus allen Berechnungsblättern der alten Berechnung
`\${Spalte}_final.csv	eine CSV-Datei mit den Werten der jeweiligen Spalte aus allen Berechnungsblättern der neuen Berechnung
`\${Spalte}_diff.csv	eine CSV-Datei mit der nominalen Abweichung der neuen Berechnung von der alten Berechnung
`\${Spalte}_diff_percent.csv	eine CSV-Datei mit der prozentualen Abweichung der neuen Berechnung von der alten Berechnung
`\${Spalte}_diff_percent.html	eine HTML-Datei, die die prozentuale Abweichung der neuen Berechnung von der alten Berechnung tabellarisch darstellt, jeweils unterlegt mit einer Hintergrundfarbe, die anzeigt, wie stark die Abweichung ausfällt – siehe Abbildung 13

Abbildung 13: Ausschnitt aus IN_diff_percent.html

	IN-1995	IN-1996	IN-1997	IN-1998	IN-1999	IN-2000	IN-2001	IN-2002	IN-2003	IN-2004	IN-2005	IN-2006	IN-2007	IN-2008	IN-2009
Region-SNR															
DG	0.000000	0.000000	-0.000000	0.000000	0.000000	-0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	-0.000000
01	0.000000	0.000000	0.000000	-0.000000	-0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	-0.000000	0.000000
02															
03	-0.000000	0.000000	0.000000	0.000000	-0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
04															
05	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	-0.000000	0.000000	0.000000	0.000000	-0.000000	-0.000000
06	0.000000	0.000000	-0.000000	-0.000000	0.000000	0.000000	-0.000000	0.000000	0.000000	0.000000	0.000000	-0.000000	-0.000000	0.000000	-0.000000
07	0.000000	0.000000	-0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	-0.000000
08	-0.000000	0.000000	0.000000	0.000000	0.000000	-0.000000	0.000000	0.000000	0.000000	0.000000	-0.000000	0.000000	0.000000	0.000000	0.000000
09	0.000000	-0.000000	-0.000000	-0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	-0.000000	0.000000
10	0.000000	0.000000	0.000000	-0.000000	0.000000	0.000000	0.000000	0.000000	-0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	-0.000000
11															
12	0.000000	0.000000	-0.000000	-0.000000	0.000000	0.000000	-0.000000	-0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	-0.000000	0.000000
13	0.000000	0.000000	-0.000000	0.000000	0.000000	-0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
14	0.000000	0.000000	-0.000000	-0.000000	0.000000	0.000000	-0.000000	0.000000	-0.000000	0.000000	0.000000	0.000000	0.000000	-0.000000	0.000000
15	0.000000	-0.000000	-0.000000	-0.000000	0.000000	-0.000000	-0.000000	0.000000	0.000000	0.000000	-0.000000	0.000000	0.000000	0.000000	-0.000000
16	0.000000	0.000000	-0.000000	0.000000	-0.000000	0.000000	-0.000000	0.000000	0.000000	0.000000	-0.000000	-0.000000	0.000000	0.000000	-0.000000
17	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	-0.000000	0.000000
010															
020															

Dargestellt ist ein typisches Beispiel für einen quantitativen Vergleich ohne gefundene Fehler. Die konkrete Darstellung zeigt die prozentuale Abweichung des Ergebnisses der Neuimplementierung vom Ergebnis der korrigierten Referenzimplementierung für die Spalte IN (also die Bilanzsumme der N-Zufuhr). Farblich dargestellt ist die Höhe der Abweichung vom Referenzwert. Die dargestellten Kategorien sind: **grün** (keine Abweichung), **gelbgrün** (Abweichung <= 0,01 %) – die Kategorien **gelb** (Abweichung <= 0,1 %), **orange** (Abweichung <= 1%) und **rot** (Abweichung > 1 %) treten in der Beispielabbildung nicht auf. Die gefundenen Abweichungen bewegen sich im Maß der erwarteten Unsicherheiten für Fließkommazahlen und sind damit unerheblich.

Quelle: eigene Darstellung, PIKobytes GmbH

3.6 Entwicklung von Werkzeugen zur Analyse der Datenvorverarbeitung

Mit der Fertigstellung der Arbeiten an den Kernberechnungen zur N-Flächenbilanzierung wurden weitere Entwicklungsarbeiten im Bereich der Datenvorverarbeitung durchgeführt. Wie bereits in Kapitel 3.2.2 beschrieben, sind die Vorberechnungen in weiteren Excel-Dateien vorhanden, die unabhängig von den Kernberechnungen analysiert und migriert werden müssen.

Diese Arbeiten haben vorläufigen Charakter, da während der Projektlaufzeit nicht abschließend geklärt werden konnte, von welchen Stellen die Eingangsdaten künftig bezogen werden sollen. Während ein großer Teil des statistischen Mengengerüsts weiterhin aus der Officialstatistik stammen wird, laufen parallel dazu Abstimmungsprozesse mit dem Thünen-Institut hinsichtlich der Nutzung einer gemeinsamen Datengrundlage für die Stickstoffbilanzierung.

Im Zuge dieses Projektes wurde festgelegt, die Entwicklungen zur Datenvorverarbeitung wie folgt einzuschränken:

► **Umfang:**

- Die Vorberechnungen werden für Eingangsdaten der Kern-Berechnung analysiert und neu implementiert, dazu gehören
 - Agrarflächen
 - Tierbestände
 - Erträge
- Die Vorverarbeitung folgender Parameter der Kernberechnung wird zunächst nicht analysiert
 - Milchleistung (kann eventuell dennoch mit wenig Aufwand umgesetzt werden)
 - Biomasse- und Biomethananlagen
 - Inventardatenaufbereitung
 - Substratdaten

► **Tiefe:** Die Arbeiten an den Vorberechnungen umfassen

- Lückenschließung
- Ersatzberechnung von Rohdaten (meist auf Kreisebene) aus anderen Rohdaten (z. B. Weizenanbaufläche der Kreise aus Weizenanbaufläche der Bundesländer und Winterweizen für beide)
- Gebietsumlegung der Kreise auf Kreisregionen der Kernberechnung
- Es werden lediglich direkte Eingangsdaten der Kern-Berechnung berücksichtigt. Zusätzliche Stufen in der Vorverarbeitung (im Sinne von Vor-Vorberechnungen) werden zunächst nicht analysiert.

3.6.1 Analyse von Vorberechnungsdateien

Voraussetzung für die Re-Implementierung der Vorberechnungen ist, wie bei der Kernberechnung, die Analyse und Dokumentation der entsprechenden Rechenvorschriften aus

den Excel-Rechenmappen für die Datenaufbereitung. Folgende Vorberechnungsdateien wurden analysiert und dokumentiert:

- ▶ Akt_Ertrage_Ackerland_B_L_K_2022.xlsx
- ▶ Akt_Ertrage_Körnermais_Grünland_B_L_K_2022.xlsx
- ▶ InterpolationFlächen_Grünland_Ackerland_B_L.xlsx
- ▶ InterpolationTierbestände_B_L_K.xlsx
- ▶ GewichteteErträge.xlsx
- ▶ Akt_Tierbestände_Rinder_2009_2023_B_L_K.xlsx
- ▶ Akt_Tierbestände_Rinder_Jahr_K.xlsx

3.6.2 Entwicklung von Werkzeugen zur Extraktion von Daten

Da die Vorberechnungsdateien eine andere Struktur aufweisen als die Referenzimplementierung der Kernberechnung, wurden dafür weitere individuelle Werkzeuge zur Extraktion von Daten entwickelt. Das Python-Modul *extract_raw_data_utils.py* stellt dafür generische wiederverwendbare Funktionalität bereit. Andere Skripte wie zum Beispiel *extract_raw_data_from_interpolation_files.py* oder *extract_raw_data_from_yac_files.py* basieren darauf und ergänzen jeweils die Details für die jeweilige Quelldatei.

3.7 Entwicklung von Rechenmodulen zur Datenvorverarbeitung

Die zusätzlichen bzw. vorgelagerten Eingangsdaten aus der Officialstatistik liegen zunächst noch nicht in der für die Kernberechnung erforderlichen Form vor. Die verfügbaren Daten werden teilweise nicht jährlich fortgeschrieben, sondern nur alle zwei, drei oder vier Jahre erhoben. Dadurch können Zeitreihen Datenlücken aufweisen. Es kommt auch vor, dass für einzelne Kreisregionen keine Daten existieren. Um die Datenlücken zu schließen, müssen die Rohdaten zunächst durch einschlägige Verfahren inter- bzw. extrapoliert werden. Außerdem ist eine Umrechnung der Daten von den Gebietszuschnitten der Officialstatistik auf die in der N-Flächenbilanzierung verwendeten Kreisregionen notwendig.

Teilweise liegen auch die Rohdaten für die Interpolation / Regionsumrechnung nicht in der benötigten Form vor oder müssen erst aus anderen Größen berechnet werden. Dafür sind weitere Vorberechnungen erforderlich.

Bei der Neuimplementierung dieser Vorberechnungen in Python gibt es verschiedene Umsetzungswege für die verschiedenen Arten der Vorberechnung.

Einige Vorberechnungsdateien sind ähnlich aufgebaut wie die Excel-Dateien für die Kernberechnung. In diesem Fall werden die zu berechnenden Größen bzw. Eingangswerte in den Spalten und die Kreisregionen in den Zeilen geführt. Die Berechnung erfolgt dann jeweils für ein Jahr, d. h. es gibt für jedes Jahr eine Berechnungstabelle. Hier können die Vorberechnungen auch in Python nach dem gleichen Prinzip wie die Kernberechnung implementiert werden, zum Beispiel im Modul *pre_calculation_y_ac.py* für die landwirtschaftlichen Erträge.

Bei anderen Rechenmappen der Datenvorverarbeitung werden die einzelnen Jahre hingegen in den Spalten geführt und die Kreisregionen in den Zeilen. Jede berechnete Größe wird in einer

eigenen Tabelle geführt. Für diese abweichende Struktur kann dann die Herangehensweise aus der Kernberechnung nicht übernommen werden. Dieses ist zum Beispiel bei der Interpolation/Extrapolation (also dem Schließen von Datenlücken in einer Zeitreihe) und der Umrechnung auf die in der Stickstoffflächenbilanz verwendeten Kreisregionen der Fall. Dafür wurden neue Werkzeuge in den Python-Modulen *district_reallocation.py* und *interpolation_areas_and_livestock.py* bereitgestellt.

Die lückengefüllten und umgerechneten Daten aus der Vorverarbeitung stellen die Eingangsdaten für die Kernberechnung dar. Sie müssen von der *Jahr-Region-pro-Größe*-Form in die *Größe-Region-pro-Jahr*-Form umgewandelt werden. Diese Umwandlung erfolgt im Modul *data_transformation.py*.

Weitere Vorberechnungsdateien enthalten sehr spezifische Berechnungen und erfordern daher ebenfalls Python-Implementierungen mit individuellen Lösungen, wie zum Beispiel das Modul *pre_calculation_nctl.py*, welches die Berechnungen aus *Akt_Tierbestände_Rinder_2009_2023_B_L_K.xlsx* nachimplementiert.

3.7.1 Abfrage von Rohdaten

Einige Quelldaten für die Vorberechnungen können direkt über die API von Destatis.de (<https://www-genesis.destatis.de/genesisWS/rest/2020>) bzw. Regionalstatistik.de (<https://www.regionalstatistik.de/genesisws/rest/2020>) abgefragt werden. Dafür wurde das Python-Modul *genesis_api_client.py* entwickelt.

Weitere Quelldaten werden im Moment noch auf umständlichen Wegen beschafft, so dass eine manuelle Übertragung (Medienbruch) notwendig ist. Dadurch ist eine komplett automatische Beschaffung der Eingangsdaten derzeit nicht möglich.

4 Projektergebnisse

4.1 Softwareprojekt zur Stickstoffflächenbilanzierung

Hauptergebnisse dieses Vorhabens sind die Neukonzeption und Neuimplementierung des Rechenkerns zur N-Flächenbilanzierung auf landwirtschaftlichen Flächen auf Ebene der Kreise. Das entsprechende Softwareprojekt wurde in Kapitel 3.4 vorgestellt. Das Softwareprojekt wird in funktionsfähigem Zustand bereitgestellt. Es handelt sich um ein Werkzeug für Experten*Expertinnen, das Kenntnisse in der Programmiersprache Python voraussetzt. Die Anforderungen an die Laufzeitumgebung sowie Details zur Strukturierung der Datenverzeichnisse sind in den Kapiteln 3.4.2 und 3.4.4 beschrieben.

4.2 Validierung und technische Dokumentation der Referenzimplementierung des Modells zur Stickstoffbilanzierung

Durch die der Re-Implementierung vorausgehende Analyse und Korrektur der Referenzimplementierung (siehe Kapitel 3.3.3) wurde die bisher verwendete Berechnung erstmals ausführlich detailliert technisch dokumentiert, auf interne Fehler geprüft und berichtigt. Entsprechende Korrekturen sind auf die Excel-basierte Fortschreibung der Berechnung der Stickstoffflächenbilanz durch die AG Häußermann und Bach bereits im Jahr 2024 für die Zeitreihe bis zum Jahr 2022 übernommen worden.

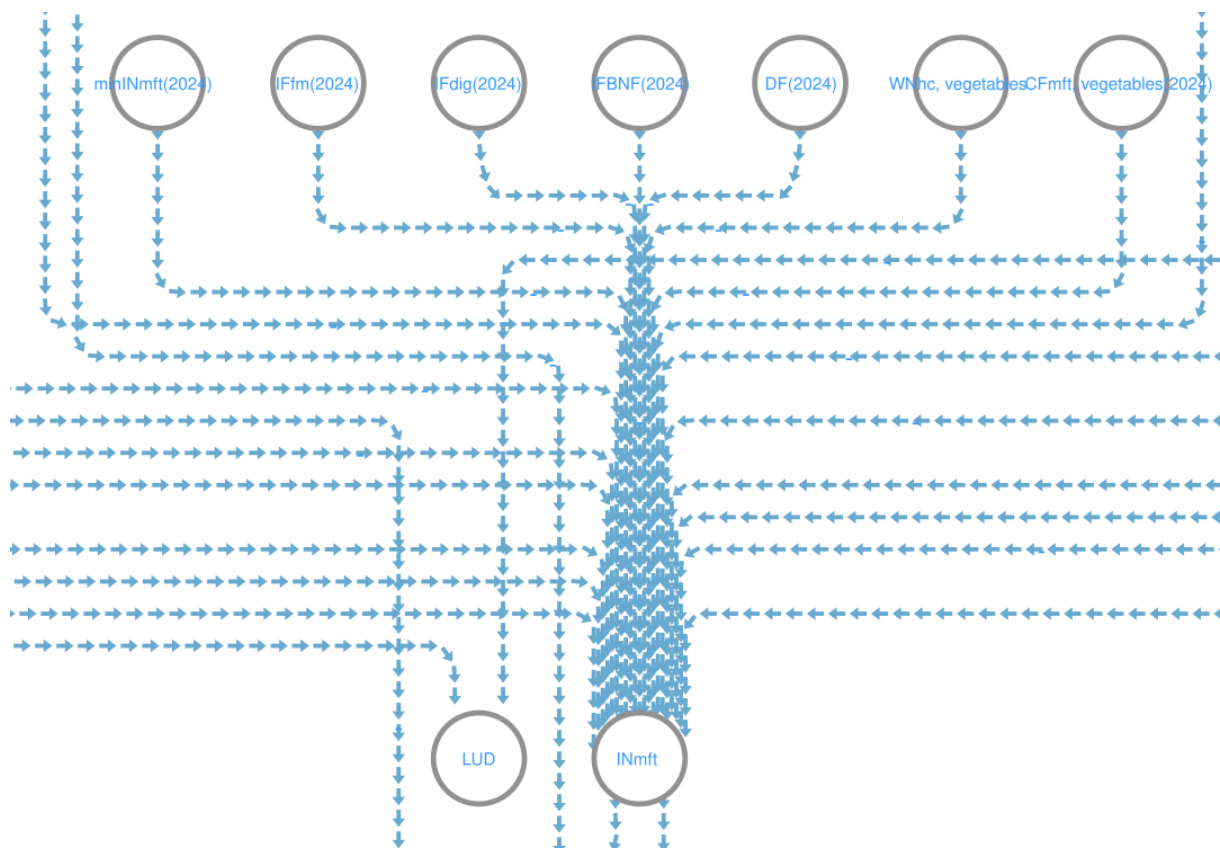
4.3 Fließschema

Mit Hilfe des in Kapitel 3.3.4 vorgestellten Werkzeugs kann ein Graph, der Abhängigkeiten zwischen den einzelnen Eingangsdaten und Zwischenergebnissen der Stickstoffflächenbilanzberechnung widerspiegelt, als GraphML-Datei ausgegeben werden.

Dieser Graph kann mit Hilfe von Werkzeugen wie Cytoscape¹² grafisch dargestellt werden (Fließschema). Dieses Fließschema wurde als Vektorgrafik exportiert und in Form der Datei *pflast_network.graphml_1.svg* am 24. Oktober 2024 an das UBA übergeben.

¹² <https://cytoscape.org/>

Abbildung 14: Fließschema (Ausschnitt)



Gezeigt wird ein Ausschnitt des baumförmigen gerichteten Graphen der Stickstoffflächenbilanzierung. Im Ausschnitt enthalten ist der Knoten IN_{mft} und ein Teil seiner Eingangsdaten (sowie der Knoten LUD, dessen Eingangsdaten außerhalb des Bildausschnitts liegen). Die gerichteten Beziehungen zwischen den Knoten sind durch Pfeile dargestellt.

Quelle: eigene Darstellung, PIKOBYTES GmbH

4.4 Funktionen und Werkzeuge zur Datenvorverarbeitung

Wie in 3.7 beschrieben sind bereits einige Skripte für die Datenvorverarbeitung implementiert. Diese umfassen in der Breite die Vorberechnungen für die Agrarflächen, die Agrarerträge sowie die Tierbestände. Das entspricht somit den wesentlichen Eingangsdaten der Kernberechnung. Weitere bislang nicht umgesetzte Vorberechnungen sind vor allem die Anbindung der Daten zur Biogasgewinnung, zur Milchleistung sowie weiterer Größen, die in die Kernberechnung als Parameter eingehen.

Die Verarbeitungstiefe der Vorverarbeitungswerkzeuge ist eingeschränkt und umfasst bisher nur die Lückenschließung und die Umlegung auf die im Bilanzierungsmodell definierten Kreisregionen. Diese Verarbeitungstiefe ist, rückschreitend vom Rechenkern aus betrachtet, nur der unmittelbar vorgelagerte letzte Schritt der Datenaufbereitung. Weiter vorausgehende Transformationsschritte in Richtung der originalen Quelldaten wurden vorerst nicht umgesetzt.

5 Weiterentwicklung und funktionale Erweiterungen

5.1 Notwendige Schritte zur Vervollständigung der Neuimplementierung

5.1.1 Datenvorverarbeitung

Größere Teile der Datenvorverarbeitung konnten im Projekt nur teilweise analysiert und neu implementiert werden. Die vollständige Ablösung der Excel-basierten Vorberechnungen ist eine wichtige Voraussetzung für den robusten und effizienten Einsatz des Bilanzierungsmodells und sollte möglich zeitnah in Angriff genommen werden. Erst nach der Ablösung dieser Vorverarbeitungsschritte durch eine nachvollziehbare und entsprechend dokumentierte Neuimplementierung ist eine geordnete Weiterentwicklung des N-Bilanzierungsmodells möglich.

5.1.2 Abschließende Prüfung der primären Datenquellen

Nach dem Abschluss der notwendigen Arbeiten an der Datenvorverarbeitung sollte eine abschließende Dokumentation aller primären Datenquellen des statistischen Mengengerüsts stattfinden. In diesem Zuge sind dann auch die Nutzungsbedingungen für sämtliche Quelldaten festzustellen, sodass klar ist, ob das Bilanzierungsmodell zusammen mit den Vorberechnungen und den erforderlichen Eingangsdaten frei veröffentlicht und weitergegeben werden darf.

Weiterhin kann in diesem Zuge erneut geprüft werden, ob künftig eine stärkere Automatisierung bei der Fortschreibung der Eingangsdaten möglich ist. In diese Entscheidung spielen allerdings zusätzlich Aspekte zur langfristigen Stabilität der externen Schnittstellen und Zusicherungen der Datenbereitsteller*innen über die künftige Verfügbarkeit gleichartiger Daten eine wichtige Rolle. Sollten hier Zweifel bestehen, ist ein turnusmäßiger manueller Datenabruf, verbunden mit einer manuellen Datenprüfung weiterhin zu bevorzugen.

5.2 Verstetigung der Modellpflege am UBA

Mit der Übergabe sämtlicher Quellcodes und Dokumentationen liegt die Modellpflege in der Verantwortung des UBA. Der Modellbetrieb und die zu erwartenden Aufgaben in der Modellpflege und -fortschreibung sind inhaltlich anspruchsvoll. Sie erfordern tieferes Wissen zur Stickstoffbilanzierung, zu den benötigten bzw. verfügbaren Eingangsdaten sowie technisches Wissen und Programmierkenntnisse zur Bedienung und Modifikation des Rechenkerns. Während Entwicklungstätigkeiten zumindest teilweise ausgelagert werden können, ist eine inhaltliche und wissenschaftliche Betreuung am UBA unerlässlich.

Organisatorische Aufgaben der Modellpflege umfassen vor allem das Release-Management und die Verwaltung bzw. Ablage von Modellvarianten. Als technisches Hilfsmittel wurde im Konzept eine Versionsverwaltung empfohlen. Um diese Versionsverwaltung sinnvoll zu nutzen, müssen zusätzlich Prozesse für ein Release-Management neuer Modellversionen definiert und personell betreut werden. Für Modellvarianten ist zu entscheiden, ob diese in der gleichen Versionsverwaltung gepflegt (und damit auch reproduzierbar archiviert) werden sollen.

Perspektivisch kann auch eine Publikation des Modells als Open-Source-Software erwogen werden. Dieser Schritt ist allerdings nur sinnvoll, wenn eine langfristige Modellpflege und eine Betreuung des Softwareprojektes sichergestellt sind. Aufgrund der engen Kopplung an die Modelldaten ist in diesem Fall eine Publikation zusammen mit den erforderlichen Eingangsdaten (d. h. der vollständigen Workspaces) unbedingt zu empfehlen, sofern die Weitergabebestimmungen (vgl. Kapitel 5.1.2) der Eingangsdaten dies zulassen.

5.3 Perspektiven zur funktionalen Weiterentwicklung

5.3.1 Sensitivitätsanalysen

Im Zuge der Projektbearbeitung und Datenprüfung wurde wiederholt diskutiert, welchen quantitativen Einfluss bestimmte Eingangsgrößen und Modellparameter auf das Bilanzierungsergebnis haben. Für die Beantwortung dieser Frage haben sich in der Modellierung sogenannte Sensitivitätsanalysen etabliert. Ein vielfach genutztes Verfahren ist die Monte-Carlo-Simulation, bei der programmatisch mehrere tausend Modellläufe mit leichten, randomisierten Abweichungen in den Eingangsdaten und Parametern gerechnet und im Nachgang statistisch ausgewertet werden.

In der bisherigen Excel-Implementierung war es kaum oder nur unter sehr hohen Aufwänden möglich, solche Sensitivitätsanalysen auszuführen. Mit der Umstellung auf eine leistungsfähige Programmier- und Ausführungsumgebung ist die Durchführung derartiger Analysen deutlich leichter. Diese können künftig sogar im Zuge der Fortschreibung vollautomatisiert gerechnet werden, um die Validität der Bilanzierungsergebnisse zu belegen.

5.3.2 Geografische Auswertungen und Karten

Visuelle Darstellungen und insbesondere auch Karten zur räumlichen Verteilung der Ergebnisse der N-Flächenbilanzierung tauchen vereinzelt in Informationsangeboten des UBA auf. Diese werden anlassbezogen händisch erzeugt und dann an die Betreuer von GIS-Kartendiensten übergeben.

Mit der Umstellung auf eine leistungsfähige Programmier- und Ausführungsumgebung kann dieser Prozess vereinfacht und verstetigt werden. Die automatische Berechnung von Datengrundlagen für Kartendienste ist mit geringem Mehraufwand möglich. Jenseits der Veröffentlichung von Daten können diese Darstellungen auch für die Datenkontrolle und Plausibilitätsprüfung genutzt werden.

6 Quellenverzeichnis

Häußermann, U.; Bach, M.; Klement, L.; Breuer, L. (2019): Stickstoff-Flächenbilanzen für Deutschland mit Regionalgliederung Bundesländer und Kreise – Jahre 1995 bis 2017, UBA TEXTE 131/2019.