RepoSTAR

Ein Codepaket zur Steuerung und Auswertung statistischer Rechenläufe mit dem Programmpaket RepoTREND

GRS - 411

Gesellschaft für Anlagen-
und Reaktorsicherheit
(GRS) gGmbH

RepoSTAR

Ein Codepaket zur Steuerung und Auswertung statistischer Rechenläufe mit dem Programmpaket RepoTREND

Dirk-Alexander Becker

Mai 2016

Anmerkung:

Die diesem Bericht zugrunde liegenden Arbeiten wurden mit Mitteln des Bundesministeriums für Wirtschaft und Energie (BMWi) im Rahmen des Projektes ADEMOS unter dem Förderkennzeichen 02 E 10367 durchgeführt.

Die Arbeiten wurden von der Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH ausgeführt. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Auftragnehmer.

Der Bericht gibt die Auffassung und Meinung des Auftragnehmers wieder und muss nicht mit der Meinung des Auftraggebers übereinstimmen.

GRS - 411
ISBN 978-3-944161-93-8

Deskriptoren:
Endlager, Langzeitsicherheit, Programmcode, Sensitivitätsanalyse, Statistik, Unsicherheitsanalyse

Inhaltsverzeichnis

1 Einleitung

Das Programmpaket *RepoTREND* /REI 16/ zur integrierten Langzeitsicherheitsanalyse von Endlagersystemen ermöglicht neben der deterministischen, d. h. auf festen Eingabedaten beruhenden Untersuchung genau definierter Rechenfälle auch die Durchführung statistischer (oder probabilistischer) Analysen. Der Grundgedanke ist dabei, diejenigen Parameter, deren Werte mit signifikanten Ungewissheiten behaftet sind, durch geeignete statistische Verteilungen zu repräsentieren und in einer Vielzahl von Einzelrechenläufen zahlreiche mögliche Wertekombinationen durchzuspielen. Die so erzeugten Sätze von Ergebnissen können dann statistisch analysiert werden und erlauben auf diese Weise Rückschlüsse auf die Ungewissheit der Analyse. Dies wird als *Ungewissheitsanalyse* (oder Unsicherheitsanalyse[1]) bezeichnet. Weiterhin kann mithilfe geeigneter Verfahren die Sensitivität des Systems gegenüber Schwankungen der verschiedenen Parameterwerte ermittelt werden. Eine derartige so genannte *globale Sensitivitätsanalyse* kommt zum einen dem generellen Systemverständnis zugute, zum anderen erlaubt sie gegebenenfalls die Fokussierung weiterer Untersuchungsaktivitäten auf bestimmte Problemparameter.

Probabilistische Ungewissheits- und Sensitivitätsanalysen werden im Programmpaket *RepoTREND* über einen speziellen Statistik-Rahmen realisiert, der den Namen *RepoSTAR* trägt (*RepoTREND* framework for STAtistic Runs). Bei der Konzeption standen, insbesondere nach Erfahrungen mit früher eingesetzten Statistikwerkzeugen, Flexibilität, Benutzerfreundlichkeit und Robustheit gegen Bedienungsfehler im Vordergrund.

Der vorliegende Bericht soll eine Übersicht über die Funktionsweise des Statistikrahmens in der aktuellen Version vom 13. Mai 2016 geben und für den Benutzer eine Anwendungshilfe darstellen. Auf Zweck und Wesen der statistischen Ziehungs- und Auswerteverfahren sowie die mathematischen Hintergründe wird dabei nicht oder nur insoweit eingegangen, wie es für ein sinnvolles Verständnis der Programmfunktionalität erforderlich ist. Für tiefer gehende Informationen sei auf den Abschlussbericht zum Projekt MOSEL verwiesen /SPI 16/.

---

[1] Dieser Ausdruck wird in der deutschsprachigen Literatur häufig verwendet. Im Kontext von Sicherheitsanalysen kann er jedoch irreführend erscheinen. Deshalb wird in diesem Dokument für „unsicheres Wissen“ durchgängig der Begriff *Ungewissheit* verwendet.

Statistische Rechenläufe mit dem Programmpaket *RepoTREND* werden ebenso wie deterministische Rechenläufe über die Benutzerschnittstelle *XENIA* mit Daten versorgt, die speziell hierfür konzipierte Funktionalitäten bietet. Diese werden im vorliegenden Bericht erläutert, generell wird jedoch eine ausreichende Vertrautheit im Umgang mit *XENIA* vorausgesetzt /REI 11/, /REI 16/.

Zunächst wird in Kapitel 2 die Gesamtkonzeption dargestellt. Dem logischen Ablauf einer probabilistischen Analyse folgend wird dann im Kapitel 3 auf die Stichprobenziehung, im Kapitel 4 auf die Datenversorgung und Durchführung der Einzelrechenläufe und im Kapitel 5 auf die statistische Auswertung eingegangen. In den Anhängen werden einige spezielle mathematische und programmtechnische Konzepte erläutert, die für den Programmnutzer von Bedeutung sein können.

2 Konzeption

Zunächst wird im Folgenden das bei *RepoSTAR* verfolgte Konzept erläutert. Dabei werden auch die in diesem Bericht verwendeten Begriffe definiert.

2.1 Generelle Anmerkungen zu probabilistischen Analysen

Bei statistischen Rechenläufen wird eine große Anzahl von Einzelsimulationen mit unterschiedlichen Eingabedatensätzen durchgeführt. Diese Einzelsimulationen werden als *Spiel* bezeichnet. Die Daten werden aus einer so genannten *Stichprobe* abgeleitet. Das ist eine als repräsentativ anzusehende Menge von Wertesätzen für diejenigen Modellparameter, für die kein eindeutiger Zahlenwert angegeben werden kann oder soll. Die Zahl der Wertesätze in der Stichprobe heißt *Stichprobenumfang*. Die Werte können als (Pseudo-)Zufallsreihe oder nach anderen Prinzipien festgelegt werden. Dieser Vorgang wird als *Ziehung* bezeichnet. Dabei sind vorgegebene Verteilungen und Abhängigkeiten zu berücksichtigen.

Sinn einer probabilistischen Analyse ist zumeist, die Auswirkungen von Parameterungewissheiten auf die Modellergebnisse zu untersuchen. Bei *Ungewissheitsanalysen* interessiert nur die Gesamtwirkung aller Ungewissheiten auf das Modell, bei *Sensitivitätsanalysen* werden dagegen die Einflüsse einzelner Ungewissheiten ermittelt und miteinander verglichen. Während bei so genannten lokalen Sensitivitätsanalysen einzelne Parameter gezielt und separat im Rahmen deterministischer Rechnungen verändert und die Ergebnisse bewertet werden, werden bei einer probabilistischen, globalen Sensitivitätsanalyse alle ungewissen Parameter gleichzeitig variiert und die Ergebnisse einer speziellen Auswertung unterzogen, wodurch auch die Einflüsse von Wechselwirkungen sichtbar werden können.

Ein weiterer denkbarer Anwendungsfall für statistische Rechenläufe wäre die Einstellung von Modellparametern zur bestmöglichen Anpassung an experimentelle Ergebnisse.

Jede Parameterungewissheit wird durch eine *Verteilungsdichtefunktion* (engl.: probability density function, pdf) bestimmt, über die jedem möglichen Parameterwert eine mathematische Wahrscheinlichkeitsdichte zugeordnet wird. Die qualitative Gestalt dieser Funktion wird als *Verteilungstyp* bezeichnet. Häufig verwendete Verteilungstypen sind die (logarithmische) Gleichverteilung, die (logarithmische) Normalverteilung und die

Dreiecksverteilung. Jeder Verteilungstyp wird durch charakteristische *Verteilungsparameter* bestimmt, die das Ausmaß der Unsicherheit quantifizieren.

Verschiedene Daten, die jeweils für sich Ungewissheiten unterliegen, können z. B. aufgrund verborgener, gemeinsamer Einflüsse wechselseitige statistische Abhängigkeiten zeigen. Ein Spezialfall einer solchen Abhängigkeit ist die statistische *Korrelation*, die durch einen Korrelationskoeffizienten zwischen -1 und 1 definiert wird. Die Randwerte 1 und -1 bedeuten dabei eine strenge, d. h. durch eine direkte bzw. inverse lineare Beziehung beschriebene Abhängigkeit. Beträgt der Korrelationskoeffizient 0, dann heißen die Werte *unkorreliert*. Statistisch unabhängig verteilte Werte sind unkorreliert, die Umkehrung trifft jedoch nicht zwangsläufig zu. Zur Erläuterung sind in Abb. 2.1 verschiedene korrelierte und unkorrelierte Abhängigkeiten zweier Parameter dargestellt.

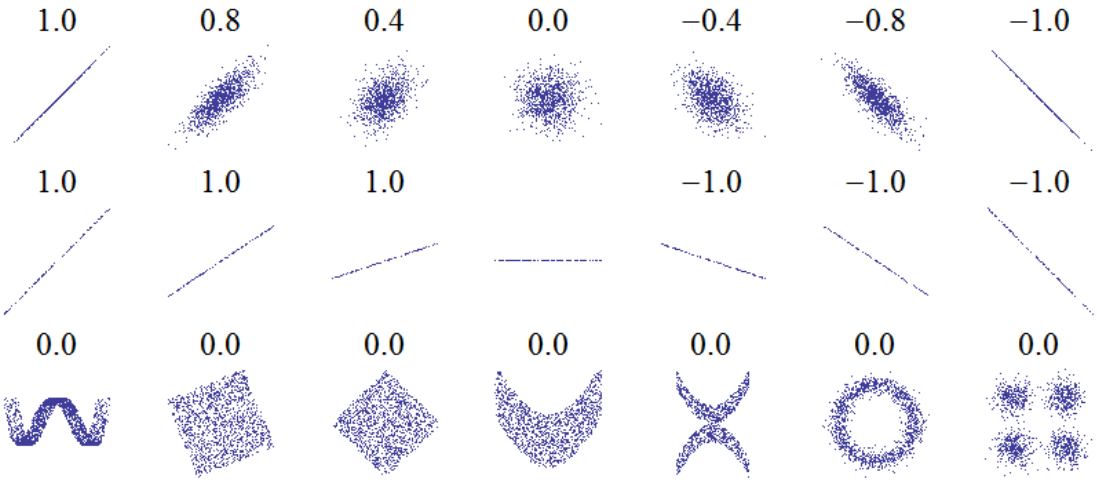


Abb. 2.1 Mögliche Verteilungen der Werte zweier Parameter mit den zugehörigen Korrelationskoeffizienten. Die Verteilungen in der unteren Zeile sind unkorreliert, obwohl sie deutliche Abhängigkeiten zeigen
(Quelle: Wikipedia)

Eine probabilistische Analyse zerfällt in drei Teile, die voneinander unabhängig ausgeführt werden:

- die Ziehung der Stichprobe,
- die automatisch gesteuerte Durchführung der Einzelsimulationen,
- die Auswertung der Ergebnisse.

Obwohl diese Teile technisch unabhängig sind, sind sie inhaltlich miteinander verknüpft, zum Beispiel erfordern bestimmte Auswertemethoden spezielle Ziehungsverfahren. Es ist daher notwendig, eine probabilistische Analyse sorgfältig zu planen.

2.2 Programmparameter, Variablen und Zuordnungsvorschriften

Eine der Hauptaufgaben von *RepoSTAR* besteht darin, die *RepoTREND*-Rechenmodule bei Beginn jedes Spiels mit den spielspezifischen Daten zu versorgen. Im Folgenden wird entsprechend der Programmkonzeption unterschieden zwischen Programmparametern und Variablen. Als *Programmparameter* werden diejenigen Größen bezeichnet, deren Werte von den Rechenmodulen eingelesen und verarbeitet werden. Der Anwender entscheidet darüber, welche Programmparameter bei dem statistischen Rechenlauf variiert werden sollen. Prinzipiell sind alle über *XENIA* definierten Eingabegrößen Programmparameter, jedoch ist nicht für alle die Möglichkeit der statistischen Variation vorgesehen.

Die zu variierenden Programmparameter werden jedoch nicht direkt in der Stichprobe festgelegt. Stattdessen definiert der Anwender einen Satz von *Variablen*, die er frei benennen und für die er Verteilungsdichtefunktionen und statistische Korrelationen festlegen kann. Ein Spiel wird durch Zuweisung eines Wertes an jede Variable definiert. Die Stichprobe umfasst dann die entsprechenden Wertesätze für alle Spiele.

Bei jedem Spiel wird aus den Werten der Variablen in der Stichprobe ein Wertesatz für die Programmparameter abgeleitet. Damit dies in korrekter Weise geschieht, muss der Benutzer entsprechende *Zuordnungsvorschriften* festlegen. Mathematisch formuliert wird über diese Vorschriften eine Abbildung des Zahlenraums der Variablen in den Zahlenraum der Programmparameter definiert. Das Verfahren ist in Abb. 2.2 veranschaulicht.

Die Zuordnungsvorschriften werden von *RepoSTAR* erst während des statistischen Rechenlaufs jeweils vor Beginn eines Spiels ausgewertet. Durch dieses Konzept wird die Stichprobe von den Programmparametern entkoppelt. Der Anwender erhält dadurch die Möglichkeit, die Variablen problemnah zu definieren und dann festzulegen, wie diese auf die Rechenmodule einwirken sollen. Auf diese Weise können nahezu beliebige Abhängigkeiten und Mehrfachwirkungen umgesetzt werden. Für jeden ungewissen Einfluss auf das System, der untersucht werden soll, ist eine Variable festzulegen. Programmparameter, die gemeinsamen Ungewissheiten unterliegen und deshalb

in gegenseitiger Abhängigkeit variiert werden sollen, können auf eine einzige Variable zurückgeführt und so einfach und flexibel aneinander gekoppelt werden.

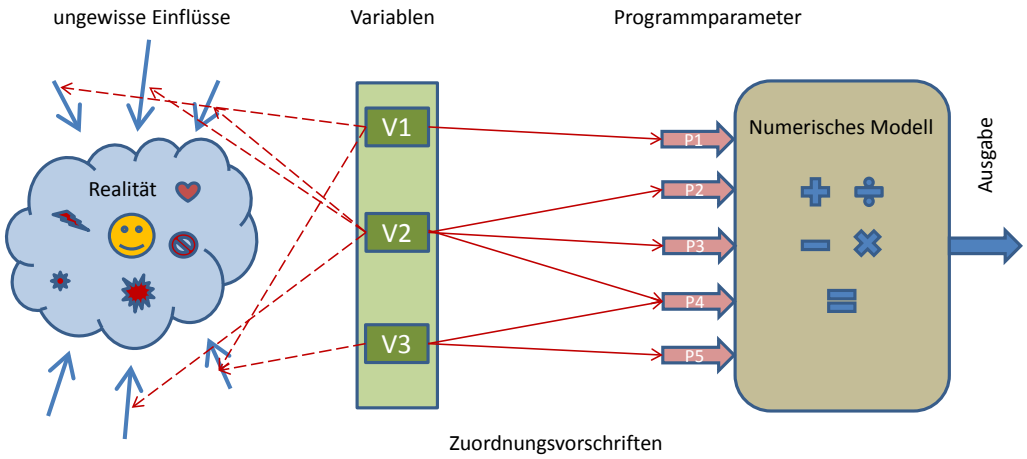


Abb. 2.2 Wiedergabe von Ungewissheiten durch Variablen, die auf die Programmparameter abgebildet werden

Bei der Planung einer probabilistischen Analyse müssen folgende Einzelheiten festgelegt werden:

- Die Variablen. Für reine Ungewissheitsanalysen können beliebig viele Variablen definiert werden. Sind jedoch Sensitivitätsanalysen durchzuführen, dann sollte generell die Zahl der Variablen nicht zu groß gewählt werden, weil sonst zu viele Spiele erforderlich werden und die Aussagekraft der Analyse abnimmt. Bei der Auswahl der Variablen sind möglichst Erfahrungen und Erwartungen sowie die beabsichtigten Auswerteverfahren zu berücksichtigen.

- Die Verteilungsfunktionen für alle Variablen. Dafür stehen verschiedene Verteilungstypen zur Verfügung. Die Verteilungstypen und -parameter können einen erheblichen Einfluss auf die Ergebnisse der Analysen haben und sollten deshalb sorgfältig gewählt werden. Eine Orientierungshilfe dafür wird in /BEC 08/ gegeben.

- Eventuell zu berücksichtigende Korrelationen zwischen den Variablen. Statistische Abhängigkeiten, die nicht als Korrelation darstellbar sind, können bei der Stichprobenziehung mit *RepoSTAR* nicht direkt berücksichtigt werden, können jedoch ggf. durch externe Manipulation der Stichprobe eingeführt werden.

- Das Ziehungsverfahren. Für reine Ungewissheitsanalysen ist eine Zufallsziehung (engl.: random) prinzipiell die beste Wahl, jedoch werden die Ergebnisse, die mit

anderen Verfahren erzielt werden, in der Praxis nur wenig davon abweichen. Für Sensitivitätsanalysen spielen andere Aspekte eine Rolle. Die robustesten Ergebnisse werden häufig mit so genannten Quasi-Zufallszahlen erreicht (engl.: quasi-random). Solche Folgen sehen auf den ersten Blick zufällig aus, werden aber tatsächlich algorithmisch im Hinblick auf eine möglichst gleichmäßige Abdeckung des Parameterraums generiert. Einige Verfahren der Sensitivitätsanalyse erfordern spezielle Ziehungsverfahren.

- Die Zahl der durchzuführenden Spiele. Für reine Ungewissheitsanalysen hängt die Mindestanzahl allein von den Genauigkeitsanforderungen an die abzuleitenden Aussagen ab. Sind Sensitivitätsanalysen geplant, dann sollte die Zahl der Variablen berücksichtigt werden. Wie viele Spiele pro Variable sinnvoll sind, hängt vom Modell, vom Ziehungsverfahren, vom Auswerteverfahren sowie von den Anforderungen an die Robustheit der Ergebnisse ab. Pauschal kann dies nicht angegeben werden, einige hundert bis wenige tausend Spiele pro Variable sind jedoch in den meisten Fällen ausreichend.

- Die Zuordnungsvorschriften, mit denen die Variablen auf die Programmparameter abgebildet werden.

- Die Zwischen- und Endergebnisse, die einer probabilistischen Analyse unterzogen werden sollen. Aus Speicherplatzgründen werden nur die ausgewählten Ergebnisse jedes Spiels für die Endauswertung aufgehoben.

- Die auf die Ergebnisse anzuwendenden Auswerteverfahren, siehe dazu /SPI 16/.

Nachdem über diese Punkte entschieden wurde, kann die Analyse in der oben genannten Reihenfolge durchgeführt werden.

2.3 Aufbau von *RepoSTAR*

Der Name *RepoSTAR* steht nicht für ein einzelnes, geschlossenes Programm oder Programmmodul, sondern für das Gesamtkonzept und seine programmtechnische Realisierung, also für die Gesamtheit der Programme und Hilfsprogramme, die für die Vorbereitung, Durchführung und Auswertung statistischer Rechenläufe mit *RepoTREND* benötigt werden. Diese Komponenten und ihr Zusammenwirken sind in Abb. 2.3 schematisch dargestellt.

Die drei Teilaufgaben „Preprocessing“, „Statistischer Rechenlauf“ und „Postprocessing“ sind voneinander getrennt und laufen unabhängig ab. Das Preprocessing kann als separater Vorbereitungs-Rechenlauf in *XENIA* definiert und vorab unabhängig ausgeführt werden oder als erstes Modul in den statistischen Rechenlauf eingebunden und somit ohne zusätzliche Benutzeraktion mit diesem zusammen ausgeführt werden. Für das Postprocessing ist diese Möglichkeit nicht vorgesehen, da dafür interaktive Benutzereingaben erforderlich sind, die im Allgemeinen erst nach Beendigung des Rechenlaufs und eventueller vorlaufender Analysen gemacht werden. Die beim Postprocessing erzeugten Daten können mithilfe des kommerziellen Programms Tecplot® oder eines Tabellenkalkulationsprogramms grafisch dargestellt werden.

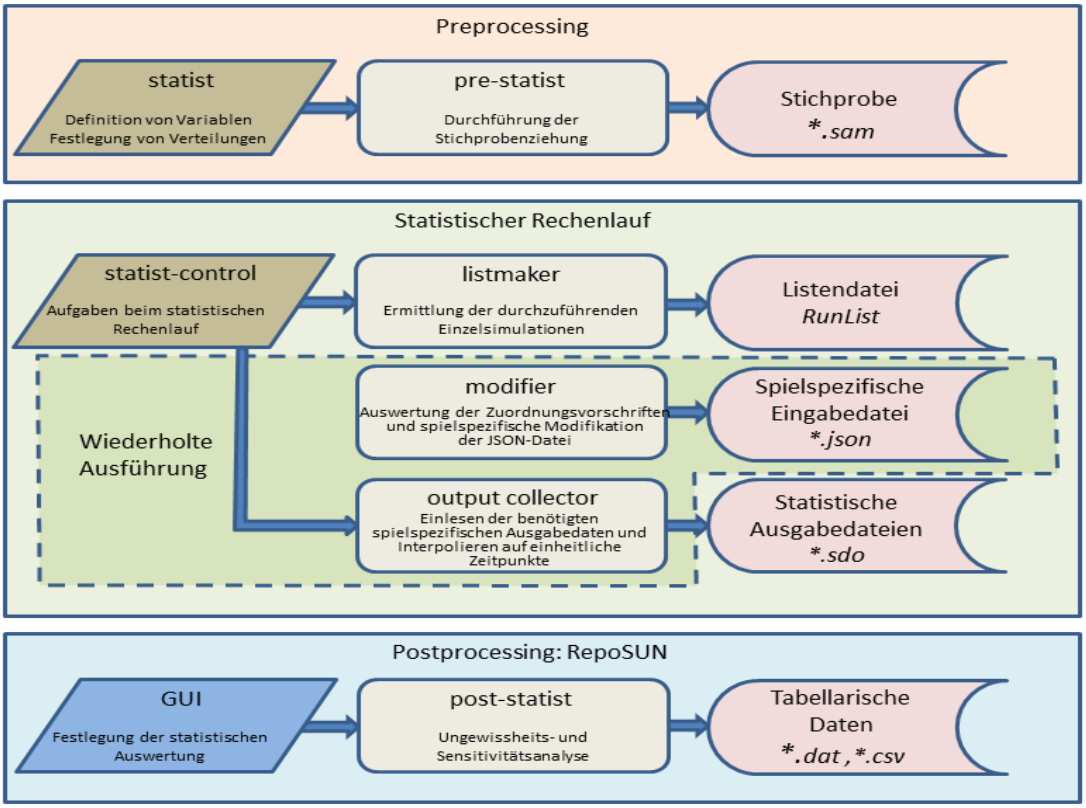


Abb. 2.3 Schematische Darstellung der Komponenten von *RepoSTAR*. Die voneinander unabhängigen Codeteile sind in der Mitte aufgeführt, die Datenversorgung erfolgt über die auf der linken Seite angegebenen *XENIA*-Module bzw. über die grafische Benutzeroberfläche (GUI) von *RepoSUN*. Die jeweils erzeugten Dateien sind auf der rechten Seite dargestellt.

3 Stichprobenziehung

Zur Ziehung der Stichprobe dient das *XENIA*-Modul *statist* (Programmname: *pre-statist*). Dieses greift auf die externe Bibliothek *SimLab4* /CER 16/ zu, welche die eigentlichen statistischen Berechnungen durchführt und dazu ihrerseits Skripte einbindet, die in der Statistik-Programmiersprache *R* /WWW-R/ verfasst sind. Zur Programmlaufzeit müssen *SimLab4*, *R* sowie die erforderlichen Skripte verfügbar sein, siehe Anhang E.

3.1 Datenversorgung

Zur Durchführung eines statistischen Rechenlaufs wird eine Stichprobendatei benötigt, in der die Werte für aller Variablen für die einzelnen Spiele festgelegt sind. Diese Datei hat die Endung *.sam, und kann mit dem Programm *pre-statist* erzeugt werden. Die Dateneingabe für das Programm *pre-statist* erfolgt über die Benutzeroberfläche *XENIA*. Dafür existiert ein *XENIA*-Modul, das den Namen *statist* trägt. Darin werden für alle Variablen Verteilungstypen und -parameter sowie ggf. Korrelationskoeffizienten festgelegt. Wenn die Stichprobendatei bereits vorhanden ist, braucht der Rechenlauf kein *statist*-Modul zu enthalten. Andernfalls kann entweder ein separater Vorbereitungs-Rechenlauf durchgeführt werden, der nur das Modul *statist* enthält, oder dieses wird als erstes Modul in den statistischen Rechenlauf eingebunden.

Es ist zu beachten, dass im Allgemeinen jede Ausführung von *pre-statist* auch bei identischen Eingaben eine veränderte Stichprobendatei erzeugt und eine eventuell vorhandene gleichnamige Datei am selben Speicherort damit überschreibt. Dadurch kann ggf. die konkrete Datengrundlage früherer statistischer Rechenläufe unwiederbringlich verlorengehen. Um dies zu verhindern kann über die Checkbox *perform sampling* eingestellt werden, ob die Ziehung zur Laufzeit tatsächlich durchzuführen ist. Wenn diese nicht markiert ist, wird *pre-statist* ohne weitere Aktion beendet und die Programmausführung wird an das nächste Modul übergeben. Ansonsten wird eine neue Stichprobendatei erzeugt.

Die zu definierenden Parameter sind in drei Knoten – general, variables, correlations – zusammengefasst, die in den folgenden Abschnitten beschrieben werden (Abb. 3.1).

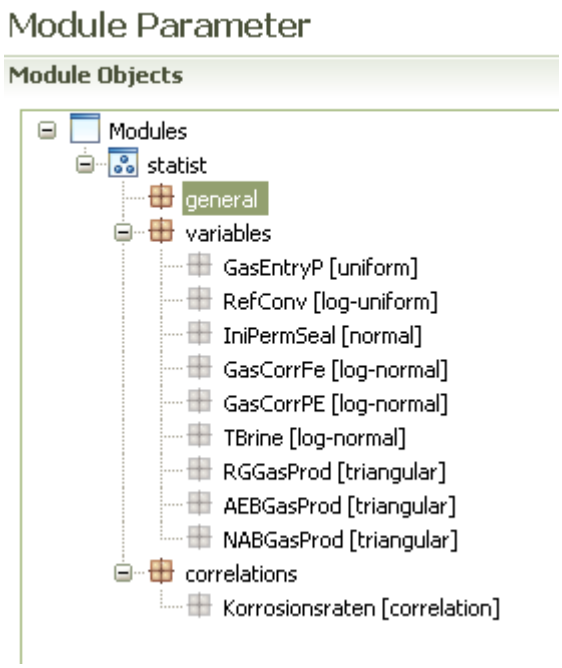


Abb. 3.1 Struktur des Moduls *statist*

3.1.1 Allgemeine Einstellungen

Unter dem Knoten *general* werden allgemeine Einstellungen für die Stichprobenziehung eingetragen (Abb. 3.2).

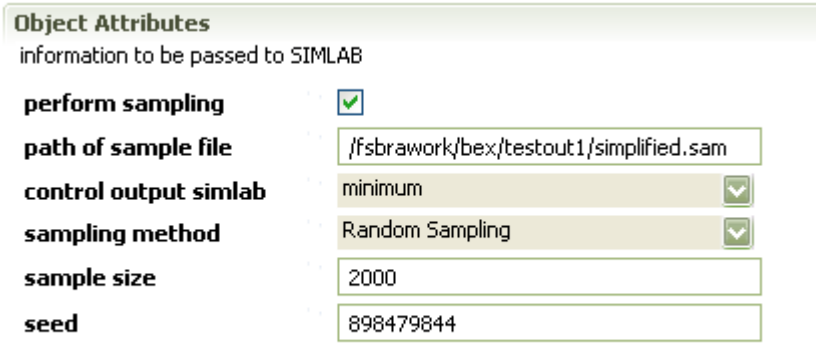


Abb. 3.2 Struktur des Knotens *general*

Die Attribute haben folgende Bedeutung:

perform sampling: Wenn die Checkbox *nicht* markiert ist, wird das Programm *prestatist* ohne Ziehung einer Stichprobe beendet. Andernfalls erfolgt eine Stichprobenziehung, und die generierte *.sam-Datei wird unter dem angegebenen Pfad abgelegt.

Falls eine gleichnamige Datei am selben Speicherort bereits existiert, so wird sie durch die neue überschrieben. Wie oben erläutert, darf nicht davon ausgegangen werden, dass die neue Datei mit der alten identisch ist, auch wenn keine Änderungen an den Eingabedaten vorgenommen wurden. Die Checkbox soll also nur dann aktiviert werden, wenn wirklich eine Neuziehung erwünscht ist.

path of sample file: Der komplette Dateipfad unter dem die zu erzeugende Stichprobendatei gespeichert werden soll. Die eingegebene Zeichenkette repräsentiert einen Pfad inklusive Dateinamen im Unix-Format. Der Dateiname sollte die Endung **.sam* haben. Falls die Datei – z. B. wegen fehlender Schreibrechte – nicht erstellt werden kann, wird eine Fehlermeldung ausgegeben.

control output simlab: Vorgesehen für spätere Programmerweiterungen, wird derzeit nicht ausgewertet.

sampling method: Auswahl des anzuwendenden Ziehungsverfahrens. Angeboten werden:

- Random Sampling,
- Latin Hypercube Sampling (LHS),
- Quasi-random LpTau,
- Classic FAST,
- Extended FAST (EFAST),
- Sobol (first and total).

Die Methoden *Classic FAST, EFAST* und *Sobol* sind speziell für die gleichnamigen Auswerteverfahren konzipiert und sollten nur verwendet werden, wenn entsprechende Auswertungen geplant sind.

sample size: Der gewünschte Stichprobenumfang. Die Zahl wird normalerweise mit der Zahl der zu rechnenden Spiele übereinstimmen. Dies ist aber nicht zwingend; bei der Durchführung des Rechenlaufs kann der Benutzer auch auf Teile der Gesamtstichprobe zurückgreifen. Bei FAST und Sobol muss jedoch die gesamte Stichprobe ausgewertet werden.

seed: Der Keim der Pseudozufallszahlenreihe. Hier sollte eine beliebige, sehr große Ganzzahl (ca. 8-10 Stellen) eingetragen werden. Dieser Eintrag wird nicht bei jedem

Ziehungsverfahren ausgewertet; ggf. arbeitet der Zufallszahlengenerator mit einem eigenen Keim.

3.1.2 Definition der Variablen

Unter dem Knoten „variables“ definiert der Benutzer die Variablen, die in die Modellrechnung einfließen sollen, mit ihren Verteilungstypen und -parametern. Für jede Variable ist ein Unterknoten einzufügen. Über den Typ des Unterknotens kann derzeit zwischen zwölf Verteilungstypen gewählt werden:

- *uniform* (Gleichverteilung),
- *log-uniform* (logarithmische Gleichverteilung),
- *normal* (Normalverteilung),
- *log-normal* (logarithmische Normalverteilung),
- *triangular* (Dreiecksverteilung),
- *Beta* (Beta-Verteilung),
- *exponential* (Exponentialverteilung),
- *Gamma* (Gamma-Verteilung),
- *Weibull* (Weibull-Verteilung)
- *multi-uniform* (Mehrintervall-Gleichverteilung, „Histogrammverteilung“),
- *multi-log-uniform* (logarithmische Mehrintervall-Gleichverteilung)
- *discrete* (diskrete Verteilung).

Die charakteristischen Parameter der Verteilung werden für jede Variable als Attribute eingetragen. Für ihre Definition sei auf die *SimLab*-Dokumentation verwiesen /SIM 04/, /CER 16/. Ein Beispiel ist in Abb. 3.3 dargestellt.

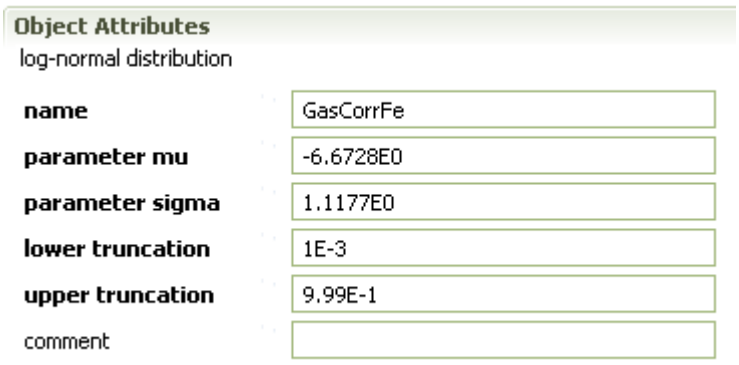


Abb. 3.3 Verteilungsparameter am Beispiel einer logarithmischen Normalverteilung

Dabei ist für jede Variable ein eindeutiger Name als Attribut „name“ festzulegen (Mehrdeutigkeit führt zu Programmabbruch und Fehlermeldung). Zur Erläuterung kann ein optionaler Kommentar *comment* eingefügt werden.

3.1.3 Korrelationen

Bei der Stichprobenziehung können Korrelationen zwischen jeweils zwei Variablen berücksichtigt werden. Diese werden unter dem Knoten „correlations“ definiert. Dort ist für jedes Paar korrelierter Variablen ein Unterknoten des Typs „correlation“ anzulegen, welcher die in Abb. 3.4 dargestellte Struktur hat.

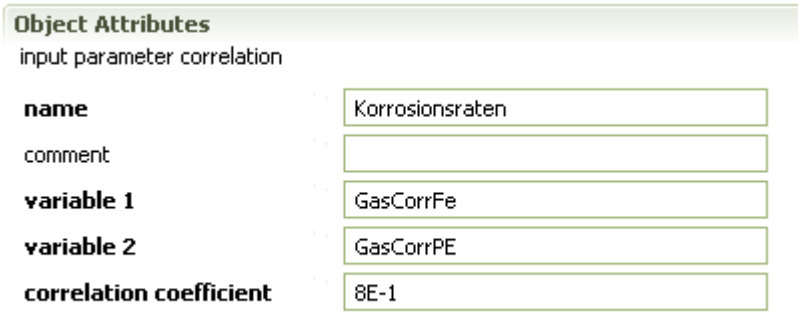


Abb. 3.4 Struktur eines Knotens vom Typ „correlation“

Der Name („name“) bezeichnet das Korrelationspaar und dient nur der Strukturierung in *XENIA*, er wird vom Programm nicht ausgewertet. Ein zusätzlicher Kommentar kann optional eingefügt werden. Die korrelierten Variablen werden über ihre Namen identifiziert, die Reihenfolge spielt dabei keine Rolle. Dem Paar wird ein Korrelationskoeffi-

zient zwischen -1 und 1 zugewiesen. Negative Werte bedeuten eine gegensinnige Tendenz, d. h. bei großen Werten der Variable 1 werden kleine Werte der Variable 2 bevorzugt gezogen und umgekehrt. Bei positivem Korrelationskoeffizienten ist die Tendenz dagegen gleichsinnig. Die Randwerte -1 und 1 ergeben den Grenzfall einer strengen (linearen) Korrelation, 0 bedeutet unkorrelierte Variablen.

Korrelationskoeffizienten werden zumeist recht willkürlich festgelegt, um eine gewisse angenommene wechselseitige Abhängigkeit abzubilden. Dafür sollten ggf. Werte im Bereich etwa zwischen 0,5 und 0,9 (absolut) verwendet werden. Korrelationskoeffizienten unter etwa 0,3 (absolut) wirken sich kaum noch erkennbar aus, sehr ausgeprägte Korrelationen sollten ggf. besser als strenge Abhängigkeit über Zuordnungsvorschriften festgelegt werden.

3.2 Ausführung des Programms

Das Programm *pre-statist* wird im Rahmen der Modulkette eines Rechenlaufs normal ausgeführt, sofern die Checkbox „perform sampling“ aktiviert ist. Andernfalls wird die Programmausführung ohne Aktion beendet.

3.3 Aufbau der Sample-Datei

Die Sample-Datei ist eine ASCII-Datei mit der Dateiendung **.sam*. Nach einigen Zeilen mit allgemeinen Informationen folgt eine dem Stichprobenumfang entsprechende Zeilenzahl mit den jeweils gezogenen Stichprobenwerten für alle Variablen. Der daran anschließende Block von Zeilen enthält die tatsächliche Korrelationsmatrix. Die folgenden Zeilen dienen der Kompatibilität mit früheren *SimLab*-Versionen und enthalten ggf. die berechneten zentralen statistischen Momente (Mittelwert, Standardabweichung, Schiefe, Kurtosis; siehe Anhang A) der Verteilungen. Diese Werte haben rein informativen Charakter und werden für die Auswertung nicht benötigt. *SimLab4* berechnet die zentralen Momente nicht, deshalb stehen hier Dummy-Werte. Im daran anschließenden Teil werden die Variablen mit ihren Namen und Soll-Verteilungen definiert. Am Ende der Datei stehen weitere Informationen, u. a. zur Soll-Korrelationsmatrix. Eine genaue Formatbeschreibung (auf Basis von *SimLab2.2*) findet sich in /SIM 04/.

4 Durchführung und Datenversorgung der Einzelsimulationen

Ein *RepoTREND*-Rechenlauf wird durch Einbindung des *XENIA*-Moduls *statist-control* als statistischer Rechenlauf gekennzeichnet. Das zugehörige Rechenprogramm ermittelt die durchzuführenden Spiele, übernimmt deren Datenversorgung und stellt die auszuwertenden Ausgabedaten zusammen. Dabei wird das Vorhandensein einer gültigen Stichprobendatei vorausgesetzt, in der die Namen und die gezogenen Werte aller Variablen für alle Spiele abgelegt sind. Diese müssen spielspezifisch auf die Programmparameter abgebildet werden. Dazu erzeugt das Programm jeweils zu Beginn der Abarbeitung eines Spiels aus der JSON-Originaldatei des Rechenlaufs eine spezifisch modifizierte JSON-Datei, die der *RepoTREND*-Modulkette dann für die Datenversorgung übergeben wird.

4.1 Eingabedaten für *statist-control*

Im *XENIA*-Modul *statist-control* werden einige Daten eingegeben. Die Struktur in *XENIA* ist in Abb. 4.1 dargestellt.

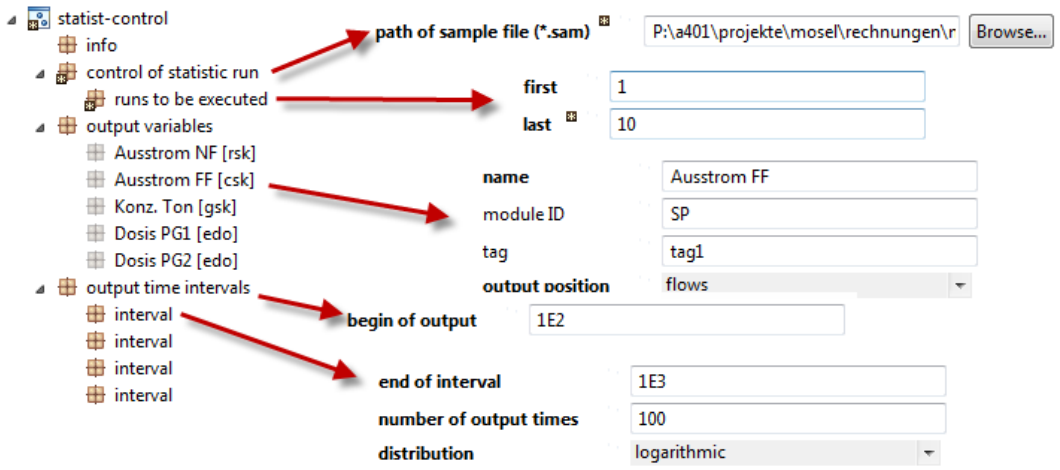


Abb. 4.1 Struktur des *XENIA*-Moduls *statist-control*. Die Pfeile verweisen auf die Attribute der Knoten.

Unter dem Knoten *control of statistic run* wird zunächst der Pfad zur Stichprobendatei eingetragen. Hierfür steht ein Dateisystem-Browser zur Verfügung. Es können beliebig viele Unterknoten des Typs *runs to be executed* eingefügt werden. Bei jedem davon werden die Nummern des ersten und des letzten zu rechnenden Spiels eingetragen.

Diese Nummern müssen selbstverständlich in der Stichprobendatei vorhanden sein. Auf diese Weise können innerhalb eines statistischen Rechenlaufs auch mehrere, nicht zusammenhängende Blöcke von Spielen in beliebiger Reihenfolge gerechnet werden.

Unter dem Knoten *output variables* werden die Ausgabedaten definiert, die von allen Spielen für die Gesamtauswertung aufbewahrt werden sollen. Diese werden aus den von den Rechenmodulen erzeugten Ausgabedateien mit den Endungen *.rsk* (erzeugt vom Nahfeldmodul *LOPOS* /HIR 99/), *.gsk* (erzeugt vom Ton-Modul *CLAYPOS*), *.csk* (erzeugt von den Modulen der *GeoTREND*-Familie /REI 16/ und *.edo* (erzeugt vom Biosphärenmodul *BioTREND* /REI 16/) entnommen.

Für jede auszugebende Größe ist ein Unterknoten des passenden Typs (*rsk*, *gsk*, *csk* oder *edo*) anzulegen. Dafür wird dann jeweils ein (frei wählbarer) Name festgelegt, weiterhin ist zu spezifizieren, welche von ggf. mehreren vorhandenen Ausgabegrößen desselben Typs gemeint ist. Dies geschieht über unterschiedliche Attribute:

Knotentyp *rsk*:
number of interface segment (optional): In *LOPOS* können mehrere Übergabesegmente definiert sein. Die hier einzutragende Nummer identifiziert das gewünschte Übergabesegment, wobei die in der **.rvs*-Datei gegebene Reihenfolge gilt. Wird nichts eingetragen, wird automatisch das erste Übergabesegment verwendet.
output position: Hier ist auszuwählen, ob Nuklidströme (flows) oder -konzentrationen (concentrations) ausgegeben werden sollen.

Knotentyp *gsk*:
tag (optional): *CLAYPOS* kann ggf. mehrere **.gsk*-Dateien erzeugen, die dann durch ein eindeutiges Kennzeichen markiert werden. Dieses ist hier einzutragen. Bleibt das Feld leer, wird ein leerer Tag angenommen.
output position: Hier ist auszuwählen, ob Nuklidströme (flows) oder -konzentrationen (concentrations) ausgegeben werden sollen.

Knotentyp *csk*:
module ID (optional): Es können ggf. mehrere *GeoTREND*-Module im Rechenlauf vorhanden sein, die jeweils eigene **.csk*-Dateien erzeugen. Diese werden über eine eindeutige, im Modul vom Benutzer festzulegende ID gekennzeichnet. Diese ist hier einzutragen. Bleibt das Feld leer, wird eine leere ID angenommen.
tag (optional): Ein *GeoTREND*-Modul kann ggf. mehrere **.csk*-Dateien erzeugen, die

dann durch ein eindeutiges Kennzeichen markiert werden. Dieses ist hier einzutragen. Bleibt das Feld leer, wird ein leerer Tag angenommen.
output position: Hier ist auszuwählen, ob Nuklidströme (flows) oder -konzentrationen (concentrations) ausgegeben werden sollen.

Knotentyp *edo*:
module ID (optional): Es können ggf. mehrere *BioTREND*-Module im Rechenlauf vorhanden sein, die jeweils eigene **.edo*-Dateien erzeugen. Diese werden über eine eindeutige, im Modul vom Benutzer festzulegende ID gekennzeichnet. Diese ist hier einzutragen. Bleibt das Feld leer, wird eine leere ID angenommen.
person group no.: Hier ist die Nummer der Personengruppe anzugeben. Dabei ist die in der **.evs*-Datei definierte Reihenfolge ausschlaggebend.

Unter dem Knoten *output time intervals* werden die Zeitpunkte definiert, für die eine Ausgabe aller ausgewählten Daten für die Statistik-Auswertung erfolgen soll. Die Ausgabedaten aller Rechenmodule werden dann auf diese Zeitpunkte interpoliert. Die Ausgabezeitpunkte werden über gleichmäßig aufgeteilte Intervalle definiert, von denen beliebig viele als Unterknoten vom Typ *interval* eingefügt werden können. Der Startzeitpunkt der Ausgabe wird übergeordnet festgelegt, dann werden jedem Intervall ein Endzeitpunkt, die Anzahl der Teilschritte sowie deren Verteilung (linear oder logarithmisch) zugewiesen, siehe Abb. 4.2. So kann es zwischen den Intervallen weder Überschneidungen noch Lücken geben. Es ist sicherzustellen, dass die Intervalle in der korrekten Reihenfolge angegeben werden, andernfalls kann es zu unvorhergesehenen Ergebnissen kommen.

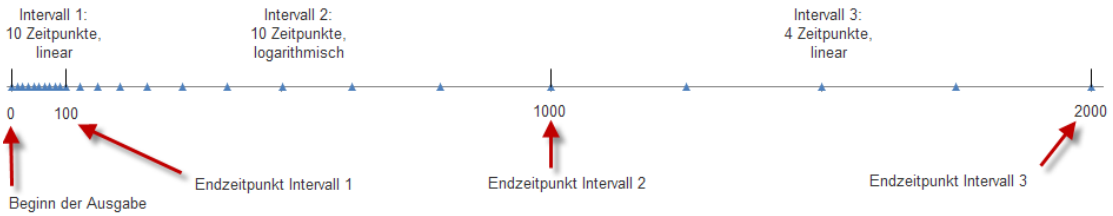


Abb. 4.2 Bestimmung der Ausgabezeitpunkte

Bei den Daten von *statist-control* können als optionale Objekte noch beliebig viele numerische Konstanten definiert werden. Auf deren Verwendung wird im Kapitel 4.3.2 eingegangen.

Für jede festgelegte Ausgabegröße wird während des statistischen Rechenlaufs eine eigene Datei mit der Endung *.sdo* erzeugt, in der die Ergebnisse aller Spiele für alle Zeitpunkte festgehalten werden. Die Benennung der Dateien setzt sich in eindeutiger Weise aus einem Buchstaben (*r*,*g*,*c* oder *e*) und den festgelegten typspezifischen Parametern zusammen.

4.2 Struktur der Modulbeschreibung

Die Hinweise in diesem Kapitel sind primär für die Entwickler von *RepoTREND*-Rechenmodulen und der zugehörigen *XENIA*-Module gedacht, können aber auch für Endanwender zum besseren Verständnis der nachfolgenden Kapitel von Interesse sein.

4.2.1 Modifikation von Programmparametern durch *statist-control*

Für die *RepoTREND*-Rechenmodule ist lediglich eine JSON-Datei relevant, welche die Programmparameter und die sonstigen Eingabedaten für den gesamten Rechenlauf enthält. Statistische Rechenläufe bestehen aus vielen Spielen, die unabhängig voneinander als Einzelrechenläufe ausgeführt werden. Jedes dieser Spiele benötigt eine eigene JSON-Datei mit den spielspezifisch veränderten Daten. Diese wird vom Programm *statist-control* automatisch erzeugt. Dazu liest es aus der **.sam*-Datei die Werte der Variablen für das jeweilige Spiel ein, wertet die vom Benutzer definierten Zuordnungsvorschriften aus (siehe Kapitel 4.3), modifiziert die in der JSON-Datei abgelegten Werte der betroffenen Programmparameter entsprechend und erstellt damit die spielspezifische JSON-Datei.

Die *RepoTREND*-Rechenmodule benötigen daher keine eigene Statistik-Steuerung. Bei der Programmierung braucht der Entwickler keine Statistik-Option zu berücksichtigen und keine Statistik-Daten einzulesen. Dennoch müssen die Zuordnungsvorschriften auf *XENIA*-Ebene in den einzelnen Modulen eingetragen werden, damit sie in der JSON-Datei an den richtigen Positionen erscheinen und von *statist-control* entsprechend verarbeitet werden können.

Der Entwickler eines *XENIA*-Moduls muss deshalb darüber entscheiden, welche Programmparameter bei statistischen Rechenläufen prinzipiell variierbar sein sollen (das bedeutet natürlich nicht, dass alle diese Daten dann auch tatsächlich variiert werden

müssen). Im Allgemeinen wird z. B. die Variation von Daten, die keine physikalischen oder logischen Eingabegrößen darstellen (z. B. Steuerparameter), wenig Sinn ergeben. Um den Anwender nicht zu verwirren, sollte eine Variationsmöglichkeit dort, wo ein Einsatz kaum zu erwarten oder nicht sinnvoll ist, auch nicht angeboten werden. Die variierbaren Programmparameter sind bei Erstellung der *XENIA*-Modulbeschreibung in besonderer Weise zu behandeln.

Zu jedem tatsächlich variierten Programmparameter wird eine Statistik-Anweisung benötigt, durch die die Zuordnungsvorschrift definiert wird. Die Eingabe dieser Anweisungen wird über spezielle Statistik-Attribute vom Typ „String“ gesteuert, die automatisch oder manuell eingefügt werden können.

4.2.2 Automatisches Einfügen von Statistik-Attributen

XENIA unterstützt das Einfügen von Statistik-Attributen in komfortabler Weise. Jedem Attribut kann in der Modulbeschreibung ein *statistic*-Flag hinzugefügt werden (Abb. 4.3). Wird dieses auf *true* gesetzt, dann erhält der Anwender ein ausklappbares Eingabefeld für die Statistik-Anweisung (Abb. 4.4).

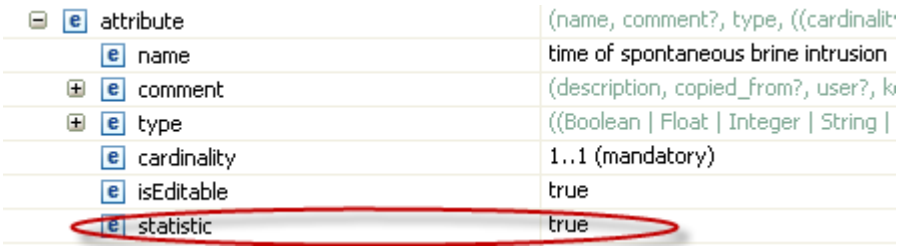


Abb. 4.3 statistic-Flag als Eigenschaft eines Attributs im Modulbeschreibungs-Editor

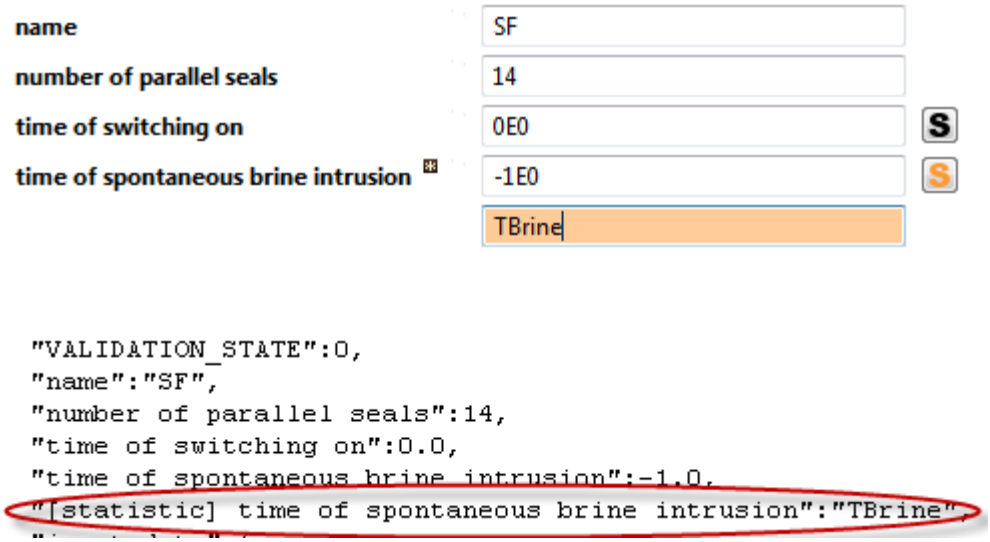


Abb. 4.4 Ausklappbares Statistik-Feld für ein Attribut. Ansicht in der Bedienungsoberfläche (oben) und Auswirkung auf die JSON-Datei (unten)

XENIA fügt dann in die JSON-Datei automatisch ein zusätzliches Attribut vom Typ „String“ auf derselben Ebene ein, das mit dem Präfix *[statistic]* gekennzeichnet und ansonsten namensgleich mit dem Hauptattribut ist. Dieser Name bleibt für den Anwender verborgen.

Es wird empfohlen, von dieser Technik Gebrauch zu machen, solange keine stichhaltigen Gründe für eine manuelle Verwaltung der Statistik-Attribute, wie sie im folgenden Kapitel beschrieben wird, vorliegen.

4.2.3 Manuelles Einfügen von Statistik-Attributen

Vorrangig aus Gründen der Kompatibilität mit älteren *XENIA*-Versionen, aber auch, um eventuell auftretenden speziellen Anforderungen gerecht zu werden, unterstützt *statist-control* die Möglichkeit, die Eingabe von Statistik-Anweisungen auch über manuell eingefügte Attribute zu realisieren. Dafür muss in der Modulbeschreibung ein optionales Attribut vom Typ „String“ eingefügt werden, dessen Name identisch mit dem des Programmparameters ist, ergänzt um das vorangestellte Kennwort *[statistic]*. Dieses Statistik-Attribut muss in der JSON-Hierarchie an einer Stelle eingefügt werden, die eine eindeutige Zuordnung zu dem Parameter erlaubt. Dafür unterstützt das Programm *statist-control* zwei Möglichkeiten:

- Für eine Statistik-Anweisung wird, sinnvollerweise unmittelbar im Anschluss an den Programmparameter, den sie modifiziert, auf derselben Hierarchieebene ein optionales Attribut eingefügt.
- Die Statistik-Anweisungen für einen Block von Programmparametern werden zusammengefasst und als optionale Attribute in einem separaten, ebenfalls optionalen Knoten (Kardinalität 0..1) untergebracht. Dieser Knoten muss mit demjenigen, der die zu modifizierenden Programmparameter enthält, namensgleich sein, ergänzt um das Präfix *[statistic]*, und sollte mit diesem als Geschwisterknoten auf derselben Hierarchieebene liegen[2] sowie sich an dessen Struktur orientieren.

Die beiden Möglichkeiten sind in Abb. 4.5 und Abb. 4.6 beispielhaft dargestellt.

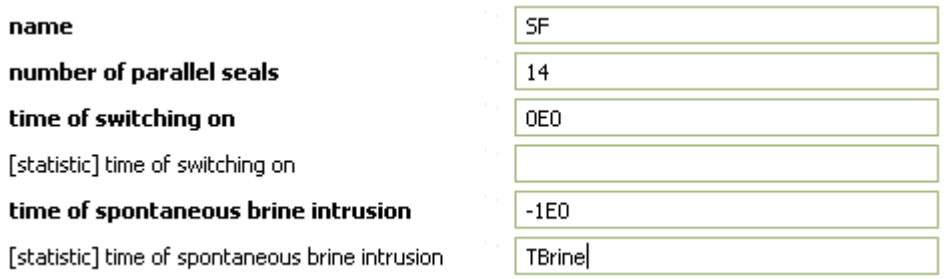


Abb. 4.5 Statistik-Anweisungen als Attribute desselben Knotens

[2] Es ist prinzipiell möglich, den Knoten mit den Statistik-Attributen auf eine andere Ebene derselben Generation auszulagern. Voraussetzung ist lediglich, dass der JSON-Pfad einer Statistik-Anweisung nach Entfernen aller *[statistic]*-Präfixe zum zugehörigen Programmparameter führt. Aus Gründen der Übersichtlichkeit und Benutzerfreundlichkeit sollten aber die Statistik-Anweisungen so nah wie möglich bei den entsprechenden Programmparametern stehen.

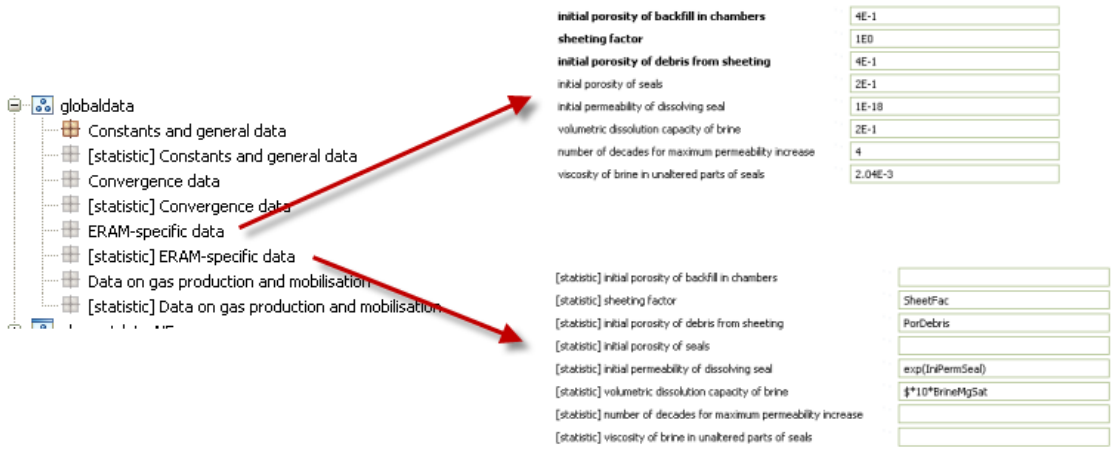


Abb. 4.6 Zusammenfassung von Statistik-Anweisungen als Attribute eines parallelen Knotens

Das Präfix *[statistic]* ist für die Identifikation von Statistik-Anweisungen reserviert und darf nur für diesen Zweck verwendet werden.

4.3 Eingabe von Statistik-Anweisungen

Über die Statistik-Anweisungen hat der Anwender die Möglichkeit, die Wirkung der gezogenen Variablen auf die Programmparameter in flexibler Weise als Zuordnungsvorschriften zu definieren. Die Anweisungen werden zur Laufzeit jeweils vor Beginn eines Spiels interpretiert. Der Wert jedes variierten Programmparameters wird dann für dieses Spiel durch einen neuen, aus der Zuordnungsvorschrift ermittelten Wert überschrieben.

Nur diejenigen Programmparameter, für die in der Modulbeschreibung die Eingabe einer Statistik-Anweisung vorgesehen wurde, können in statistischen Rechenläufen durch Statistik-Anweisungen überschrieben (variiert) werden. Sollte der Wunsch bestehen, einen Eingabewert zu variieren, bei dem kein Statistik-Feld angeboten wird, so ist der zuständige Modulentwickler zu kontaktieren.

4.3.1 Lokalisierung der Eingabefelder

Eingabefelder für Statistik-Anweisungen können in den *XENIA*-Modulen in zweierlei Art erscheinen:

- Als zusätzliches, farblich hervorgehobenes Eingabefeld, welches unter demjenigen für den zu variierenden Programmparameter erscheint, wenn auf das daneben stehende schwarze S-Symbol geklickt wird (Abb. 4.7). Bei erneutem Klick auf das (dann farbige) Symbol verschwindet das Statistik-Eingabefeld wieder.
  Achtung: Dabei geht ein eventuell vorhandener Eintrag verloren.
- Als optionales Attribut mit demselben Namen, den der zu variierende Programmparameter hat, ergänzt um das vorangestellte Präfix *[statistic]*. Diese Eingabefelder können entweder unter demselben Knoten erscheinen wie die für den Programmparameter selbst (i. d. R. direkt darunter) oder in einem dazu parallelen, speziellen Statistik-Knoten, der bei Bedarf einzufügen ist.

Abb. 4.7 Ausklappbares Eingabefeld für eine Statistik-Anweisung

4.3.2 Syntax für die Statistik-Anweisungen

Über die Statistik-Anweisungen werden die gezogenen Variablenwerte auf die Programmparameter für jedes Spiel abgebildet. Dazu dient eine einfache Formel-Syntax, die im Folgenden erläutert wird.

Die Abhängigkeit jedes zu variierenden Programmparameters von den Variablen ist als mathematische Zuweisungsgleichung festzulegen, deren rechte Seite (ohne weiteres Zuweisungszeichen) in das Feld für die Statistik-Anweisung eingetragen wird. Die Variablen werden dabei über die Namen angesprochen, die bei der Stichprobenziehung definiert wurden und in der Sample-Datei gespeichert sind. Weiterhin können die Konstanten verwendet werden, die unter *statist-control* definiert wurden. Leerzeichen zwischen Namen, Operatoren oder anderen Identifikatoren werden ignoriert.

Statistik-Anweisungen können derzeit nur für Parameter von einem numerischen Typ ausgewertet werden. Hierzu zählen auch boolesche Parameter, für die der Wert 0 als *false* und jeder andere Zahlenwert als *true* interpretiert wird. Das Ergebnis einer Statistik-Anweisung ist immer ein Fließkomma-Wert. Es ist zu beachten, dass dieser ggf. automatisch in den Datentyp des Zielparameters (Ganzzahl oder logisch) umgewandelt wird.

Im einfachsten – und häufig vorkommenden – Fall ist der Wert der Variablen direkt für den Programmparameter zu übernehmen, dazu wird einfach der Name der Variablen eingetragen.

Variablen können jedoch auch miteinander sowie mit Konstanten oder Zahlenwerten zu gültigen Ausdrücken verknüpft werden. Dafür stehen folgende Operatoren und Symbole zur Verfügung:

- Die binären Operatoren für die Grundrechenarten: `+`, `-`, `*`, `/`,
- Der unäre Minus-Operator: `-`,
- Der binäre Exponentialoperator: `^` oder `**` (synonym),
- Runde Klammern `()`, auch geschachtelt,
- Folgende Funktionen (siehe Dokumentation der Programmiersprache C):
  sin(.), cos(.), tan(.), asin(.), acos(.), atan(.), sinh(.), cosh(.), tanh(.), log(.) oder ln(.) (synonym), log10(.), exp(.), pow(.,.), sqrt(.), fabs(.), ceil(.), floor(.).
- Die Funktion scale(a,b,c,d,e), die eine „Strahlensatz-Abbildung“ eines Referenzpunktes von einem Referenzintervall in das Zielintervall vornimmt (die Argumente sind: a = Referenzpunkt, b = Zielintervall untere Grenze, c = Zielintervall obere Grenze, d = Referenzintervall untere Grenze, e = Referenzintervall obere Grenze), siehe dazu Abb. 4.8,
- Der ternäre Entscheidungsoperator: (B ? T : F), wobei B für einen zu evaluierenden booleschen Ausdruck steht, T für das numerische Ergebnis im Fall *true* und F für das Ergebnis im Fall *false* (z. B. wird (3.1 > 1.4 ? 1 : 0) zu 1 ausgewertet, die Klammern sind hier obligatorisch),
- Die Booleschen Operatoren: <, <=, >, >=, ==, != (siehe Dokumentation der Programmiersprache C),
- Das $-Symbol als Bezug auf den Originalwert des Programmparameters, der im deterministischen Fall verwendet wird,
- Die in der Eingabemaske für das Modul *statist-control* definierten Konstanten.

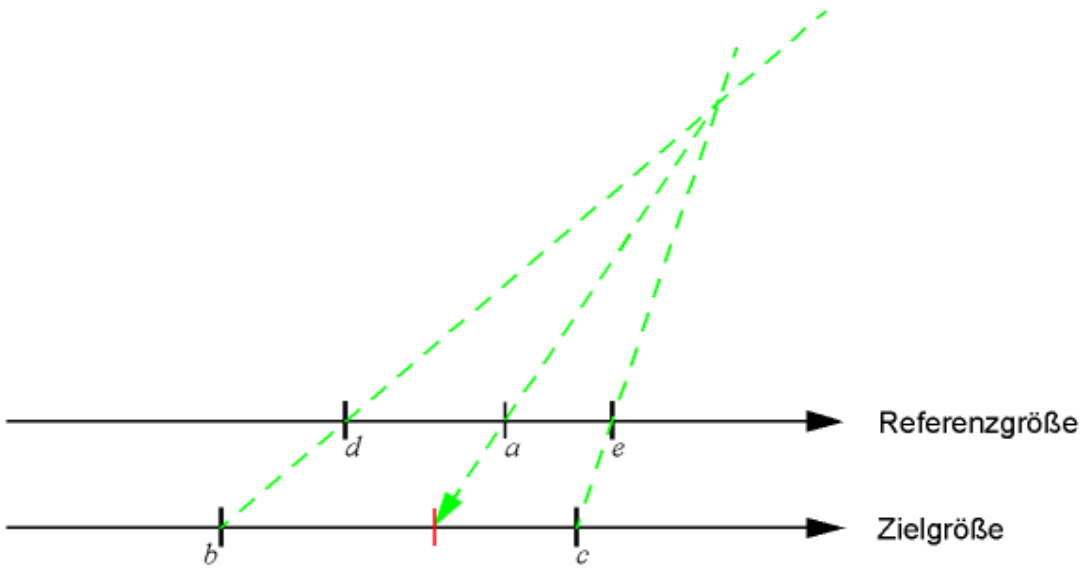


Abb. 4.8 Veranschaulichung der Funktion scale

4.3.3 Beispiele für gültige Ausdrücke

Die Syntax wird im Folgenden anhand einiger Beispiele erläutert.

Abb. 4.9 Einfache Variablen-Programmparameter-Zuordnung: der Programmparameter „gas entry pressure“, der bei deterministischen Rechnungen den Wert 2 hat, wird bei statistischen Rechnungen mit dem Wert der Variablen GasEntryP überschrieben

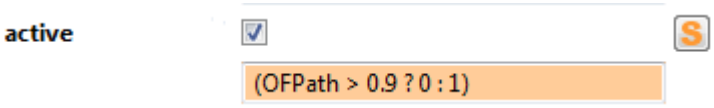


Abb. 4.10 Entscheidungs-Anweisung: Der Programmparameter „active“ vom Typ bool ist im deterministischen Fall *true* (=1). Bei statistischen Rechenläufen wird dieser Wert überschrieben durch *false* (=0), wenn die Variable OFPath größer als 0,9 ist, sonst bleibt es bei *true* (=1).

Die in Abb. 4.10 gezeigte Technik eignet sich z. B., um die Segmentstruktur im Modul *LOPOS* in Abhängigkeit von einer Variablen zu modifizieren.

Abb. 4.11 Mathematische Funktion: Hierdurch wird der Programmparameter „initial permeability of dissolving seal“ in statistischen Rechenläufen mit der Exponentialfunktion des Werts der Variablen IniPermSeal überschrieben.

Die Technik nach Abb. 4.11 kann verwendet werden, wenn z. B. ein Programmparameter mit über mehrere Größenordnungen log-normalverteiltem Wertebereich nicht direkt gezogen werden soll, sondern stattdessen der normalverteilte Exponent. Das kann erforderlich sein, weil *SimLab* bei logarithmischen Verteilungen mit weiten Bereichen teilweise Probleme bereitet.

Abb. 4.12 Ausdruck mit Bezug auf den Originalwert: Dies bewirkt, dass der Programmparameter „volumetric dissolution capacity of brine“, der im deterministischen Fall den Wert 0,2 hat, bei statistischen Rechenläufen mit dem Zehnfachen der Variable BrineMgSat multipliziert wird.

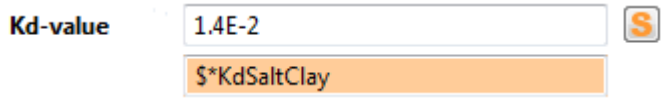


Abb. 4.13 Multiplikation des Originalwerts mit einem Faktor: Dies bewirkt, dass der elementspezifische Programmparameter „Kd-value“, der im deterministischen Fall den Wert 0,014 hat, in statistischen Rechenläufen mit dem Wert der Variable KdSaltClay multipliziert wird.

Die in Abb. 4.13 dargestellte Technik erlaubt das gekoppelte Variieren mehrerer Programmparameter über einen gemeinsamen Faktor.

5 Auswertung einer statistischen Rechnung mit dem Programm *RepoSUN*

Nach Beendigung des Rechenlaufs stehen dem Anwender Dateien mit der Endung **.sdo* für jeden Typ von ausgewählten Ausgabegrößen (*rsk, gsk, csk, edo*) zur Verfügung. Darin sind die über alle durchgeführten Spiele gesammelten Daten für alle Nuklide und alle Zeitpunkte enthalten. Da diese Dateien sehr groß sein können, sollten sie nicht mit einem Editor geöffnet werden.

Die Auswertung der Ergebnisse eines statistischen Rechenlaufs einschließlich der Erzeugung grafischer Darstellungen erfolgt nicht im Rahmen des Rechenlaufs selbst, sondern nach Abschluss desselben nach den Vorstellungen des Benutzers. Diese Auswertung wird separat und interaktiv auf der Basis der erzeugten Ergebnisse durchgeführt. Hierzu dient das Programm *RepoSUN* (**Repo**TREND **S**ensitivity and **UN**certainty analysis), welches als eigenständiges Programm mit eigener Bedienoberfläche implementiert ist. Zur Grafikerstellung erzeugt das Programm tabellarische Daten, die mittels des kommerziellen Grafikprogramms Tecplot® bzw. mit einem beliebigen Tabellenkalkulationsprogramm dargestellt werden können.

5.1 Technische Voraussetzungen

Um das Programm *RepoSUN* zu nutzen, müssen einige technische Voraussetzungen erfüllt sein, die dazu führen, dass das Programm nur unter dem Betriebssystem LINUX verwendet werden kann (getestet unter Debian7). Eine Übersicht über die benötigten Hard- und Software-Komponenten wird im Anhang E gegeben.

5.2 Hauptfenster

Nach dem Start der grafischen Benutzeroberfläche erscheint zunächst ein leeres Hauptfenster wie in Abb. 5.1 (oben) dargestellt. Über den Button *Select* erhält man einen Datei-Browser, mittels dessen eine statistische Vorspanndatei vom Typ **.svs* auszuwählen ist. Diese muss entsprechend der Beschreibung im Anhang D aufgebaut und vollständig sein. Das ist sichergestellt, wenn die Datei von einem aktuellen *RepoTREND*-Rechenlauf erzeugt wurde. Ältere statistische Vorspanndateien, die von Vorgängerversionen erzeugt wurden und bei denen die Kennwörter `%%SAMPLE`, `%%TIME` und `%%RUNS` sowie die zugehörigen Angaben fehlen, müssen gegebenenfalls

vorab von Hand bearbeitet werden. Weiterhin muss im selben Verzeichnis eine namensgleiche Datei mit der Endung *sdo* existieren, die der gültigen Spezifikation entspricht und mit der **.svs*-Datei konsistent ist. Die Stichprobendatei (**.sam*), auf die in der **.svs*-Datei verwiesen wird, muss ebenfalls den entsprechenden Formatvorschriften genügen. Da frühere Versionen von *SimLab* etwas unterschiedliche Formate erzeugt haben, ist nicht sichergestellt, dass alle älteren Stichprobendateien korrekt gelesen werden. Dies ist deshalb sorgfältig zu prüfen.

Wenn alle benötigten Dateien gefunden wurden und gelesen werden konnten, erscheinen im Hauptfenster zusätzliche Angaben, siehe Abb. 5.1 (unten). Diese Angaben umfassen neben den Dateipfaden das Ziehungsschema, den Stichprobenumfang, die Nummern des ersten und letzten in den Daten vorhandenen Einzelrechenlaufs sowie die tatsächliche Gesamtanzahl der Rechenlaufergebnisse sowie die Anzahlen der vorhandenen und ausgewählten Auswertezeitpunkte, Nuklide und Nuklidsummen.

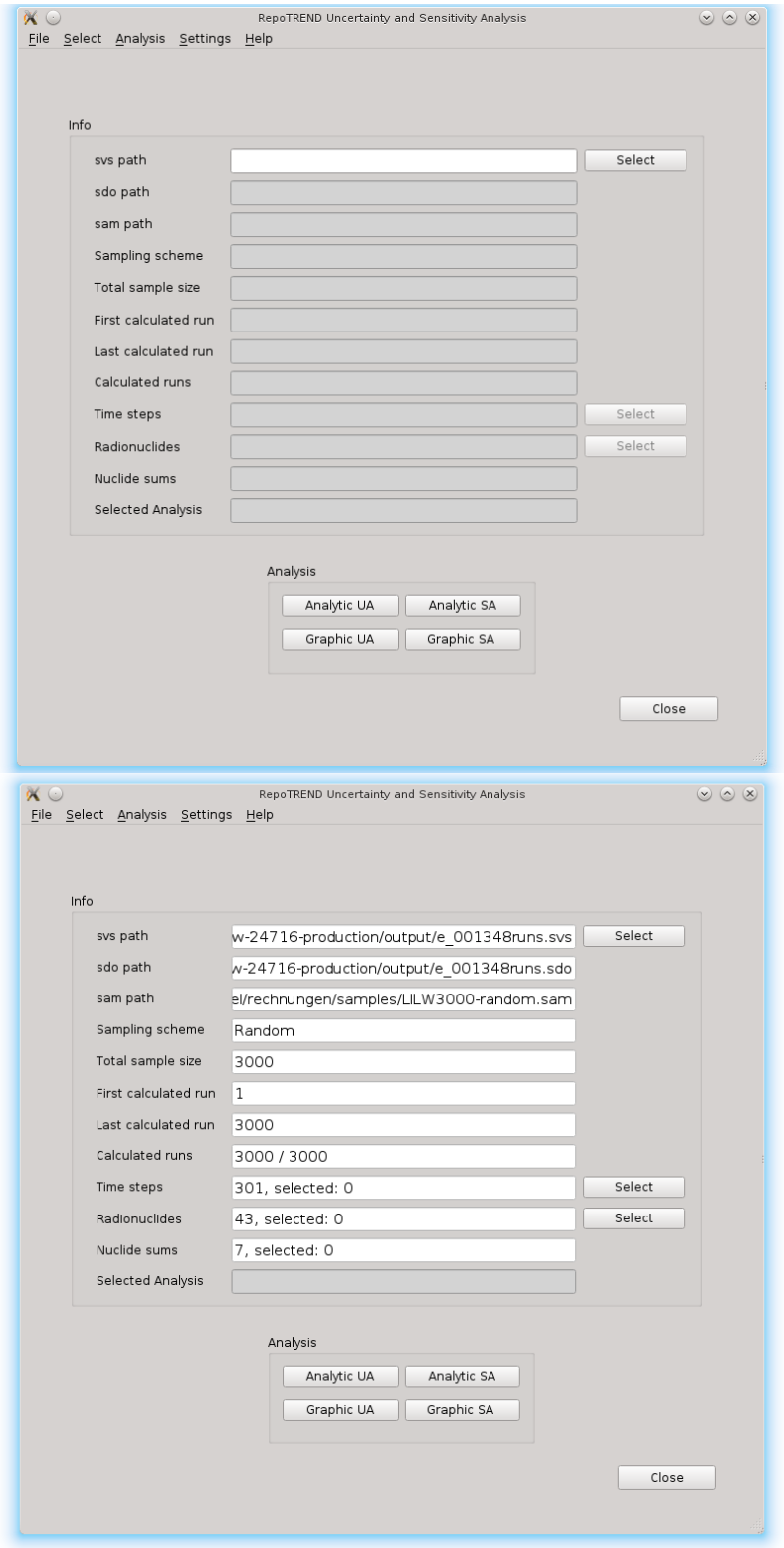


Abb. 5.1 Hauptfenster der *RepoSUN*-Oberfläche: leer (oben) und ausgefüllt (unten)

5.3 Auswahl von Zeitpunkten und Radionukliden

Wenn konsistente Daten vorgefunden wurden, werden die Buttons zur Auswahl von Auswertezeitpunkten und Radionukliden freigegeben. Diese öffnen die in Abb. 5.2 abgebildeten Unterfenster.

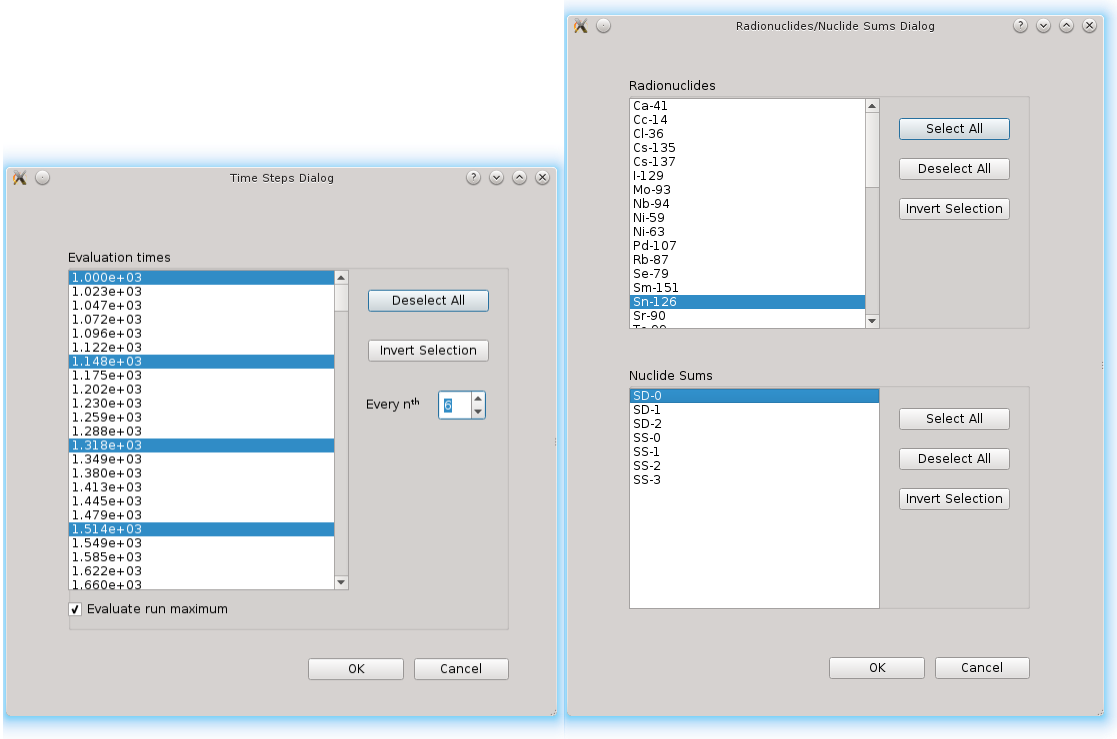


Abb. 5.2 Auswahlfenster für Auswertezeitpunkte (links) bzw. Radionuklide und Nuklidsummen (rechts)

5.3.1 Zeitpunkte für die Auswertung

Mithilfe dieses Dialogs werden von den Zeitpunkten, für die Daten vorliegen, diejenigen ausgewählt, für die der Benutzer eine probabilistische Auswertung durchführen will. Hier sollte eine sinnvolle Auswahl getroffen werden, da jeder ausgewählte Zeitpunkt zusätzlichen Rechenzeit- und Speicherbedarf verursacht.

Die Liste der Zeitpunkte wird aus der **.svs*-Datei eingelesen. Mit der linken Maustaste sowie den Tasten *Strg* und *Shift* können in üblicher Weise einzelne Zeilen oder Gruppen von Zeilen selektiert oder deselektiert werden. Mit den Buttons *Deselect All* und

Invert Selection kann die Auswahl komplett gelöscht bzw. umgekehrt werden. Über das Eingabefeld *Every nth* ist es möglich, auf einfache Weise eine gleichmäßig verteilte Teilmenge der Zeitpunkte zu selektieren. Möchte man alle auswählen, so erreicht man dies durch Eingabe einer 1. Es muss mindestens ein Zeitpunkt ausgewählt werden.

Mittels der Checkbox *Evaluate run maximum* wird erreicht, dass zusätzlich zu der Auswertung zu starren ausgewählten Zeitpunkten eine Maximum-Auswertung durchgeführt wird. Dabei werden für jeden Einzelrechenlauf Wert und Zeitpunkt des absoluten Maximums der eingelesenen Modellausgabe ermittelt. Beide Werte werden dann der Ungewissheits- oder Sensitivitätsanalyse nach den Vorgaben des Anwenders unterzogen.

Es ist zu beachten, dass die Bestimmung des Maximums nur auf die in der **.sdo*-Datei abgelegten Daten zurückgreifen kann, welche ihrerseits zeitliche Interpolationen sind. Das tatsächliche, von einem *RepoTREND*-Rechenmodul errechnete Maximum wird derzeit nicht spielspezifisch gespeichert und kann deshalb auf diese Weise auch nicht erfasst werden. In Extremfällen kann es vorkommen, dass ein scharfer Peak durch die Interpolation auf das vom Anwender definierte Zeitpunkteraster nicht erfasst und das Maximum deutlich verfehlt wird.

5.3.2 Radionuklide und Nuklidsummen

Das Dialogfenster zur Radionuklidauswahl enthält zwei Unterfenster, in denen Einzelnuklide bzw. Nuklidsummen ausgewählt werden können. In mindestens einem der beiden Bereiche muss mindestens eine Zeile ausgewählt werden. Dies geschieht in üblicher Weise mit der Maustaste und den Tasten *Shift* und *Strg*. Für das Selektieren oder Deselektieren aller Einträge sowie für das Invertieren der Selektion stehen Buttons zur Verfügung.

Die zur Auswahl stehenden Nuklide und Nuklidsummen werden aus den Einträgen in der **.svs*-Datei ermittelt, welche aus der von einem *RepoTREND*-Modul erstellten Vorspanndatei abgeleitet wird. Dementsprechend können die Einträge variieren. Üblicherweise gilt für die Nuklidsummen jedoch folgende Konvention:

SD-0: Summe über alle Radionuklide,
SD-1: Summe über die Aktivierungs- und Spaltprodukte,
SD-2: Summe über alle Zerfallsreihen,

SS-0: Summe über die 4n-Zerfallsreihe (Thorium-Reihe),
SS-1: Summe über die 4n+1-Zerfallsreihe (Plutonium-Reihe),
SS-2: Summe über die 4n+2-Zerfallsreihe (Uran-Radium-Reihe),
SS-3: Summe über die 4n+3-Zerfallsreihe (Uran-Actinium-Reihe).

Für die gewählten Nuklide und/oder Nuklidsummen wird später die Ungewissheits- oder Sensitivitätsanalyse durchgeführt. Da für jedes ausgewählte Nuklid alle Auswertungen durchgeführt werden, sollte die Auswahl im Interesse von Rechenzeit und Dateigrößen auf das tatsächlich Notwendige beschränkt werden.

5.4 Dialogfenster für verschiedene Analysen

Über die vier Buttons im Feld „Analysis“ im Hauptfenster der grafischen Benutzeroberfläche lassen sich Dialogfenster öffnen, die jeweils eine bestimmte Art der Auswertung auslösen. Diesen Dialogen sind jeweils die Buttons „Save/Compute“ und „Cancel“ sowie die Checkboxen für die Dateiformate „csv“ und „Tecplot (dat)“ gemeinsam (s. Abb. 5.3).

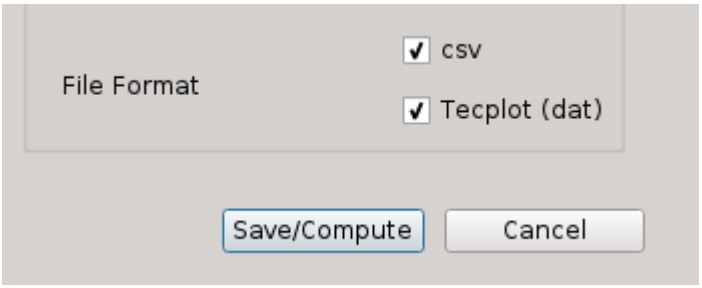


Abb. 5.3 Gemeinsame Elemente der Dialogfenster zur Auswertung

Über das Dateiformat wird festgelegt, wie die Analyseergebnisse gespeichert werden sollen. Das Dateiformat *csv* ist ein einfaches ASCII-Tabellenformat, das mit den meisten Tabellenkalkulationsprogrammen geöffnet und zur Kurvendarstellung verwendet werden kann. Das Dateiformat *dat* ist speziell auf das kommerzielle Datenvisualisierungsprogramm Tecplot® zugeschnitten. Mindestens eins der Dateiformate muss ausgewählt werden. Die Anzahl der Dateien, die das Programm erzeugt, hängt von der gewählten Auswertung ab. Darüber hinaus wird immer eine einfache Textdatei mit der Endung *.protocol* erzeugt, in der allgemeine Informationen zu den analysierten Daten sowie nicht grafisch darstellbare Ergebnisse abgelegt werden.

Der Button *Save/Compute* öffnet einen weiteren Dialog gemäß Abb. 5.4. Hier kann der Benutzer einen Basisnamen angeben, der für alle erzeugten Dateien verwendet wird, sowie einen Pfad, in dem die Dateien abgelegt werden. Der Eintrag unter „Start Script“ sollte im Normalfall nicht geändert werden. Unter „Target host“ steht eine Auswahlliste von Zielrechnern zur Verfügung. Auf dem Zielrechner müssen *SimLab4* und *R* sowie das *R*-Paket *SimLab4R* korrekt installiert sein, siehe Anhang E.

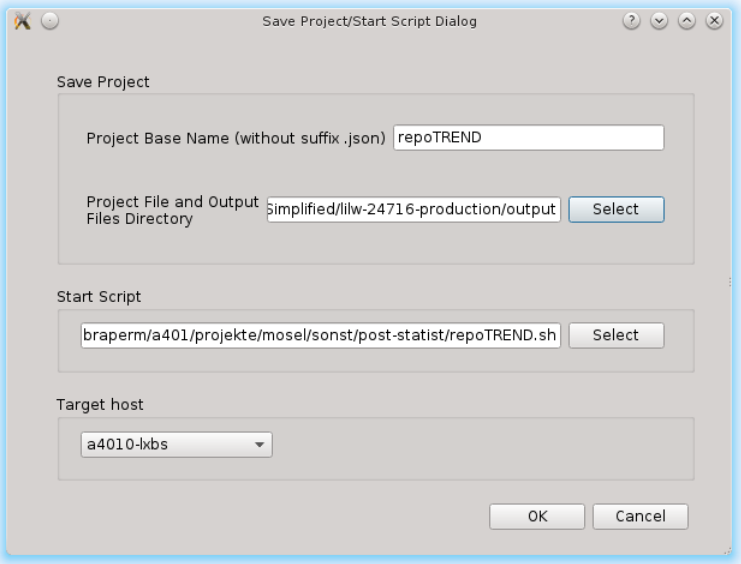


Abb. 5.4 Dialogfenster „Save/Compute“

Der Button *OK* löst die eigentliche Rechnung aus. Dazu wird zunächst eine Projektdatei mit der Endung *.json* gespeichert, die als Input für das Programm *post-statist* dient. Ist diese Datei schon vorhanden, erfolgt zuvor eine Sicherheitsabfrage.

Der Benutzer hat jederzeit die Möglichkeit, eine Projektdatei auch ohne Starten der Rechnung zu speichern. Dazu dienen die Menüpunkte *File→Save project* und *File→Save project as* im Hauptfenster. Über *File→Open project* können Projektdateien auch direkt wieder geladen werden.

Der *Cancel*-Button dient in allen Dialogfenstern zum Schließen des Dialogs, ggf. ohne dass eine Aktion ausgeführt wurde.

Im Folgenden werden die von den speziellen Analyse-Dialogen angebotenen Optionen im Einzelnen erläutert.

5.4.1 Analytische Ungewissheitsanalyse

Der Button *Analytic UA* öffnet das in Abb. 5.5 dargestellte Dialogfenster. Darin sind verschiedene statistische Maße aufgeführt. Über die entsprechenden Checkboxen sind alle Einträge vorselektiert. Die Berechnung dieser Maßzahlen erfolgt über die gesamten Modellausgabedaten für die selektierten Nuklide/Nuklidsummen und Zeitpunkte. Der Anwender kann einzelne Einträge durch Deaktivieren der Checkbox abwählen, im Normalfall besteht dafür jedoch kein Grund, da die Ungewissheitsanalyse kaum Rechenzeit beansprucht.

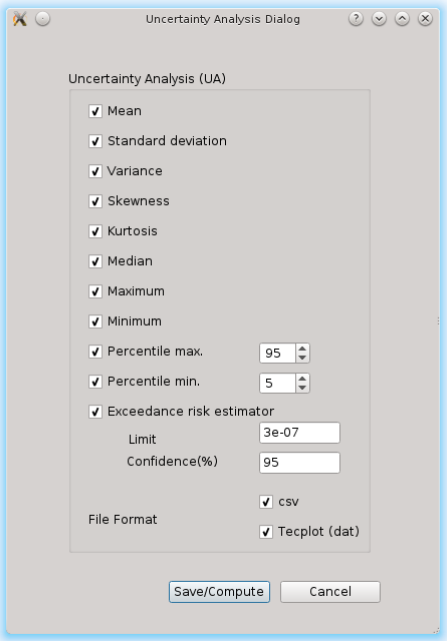


Abb. 5.5 Dialogfenster für die analytische Ungewissheitsanalyse

Die Bedeutung der Maßzahlen *Mittelwert/Erwartungswert* (*Mean*), *Varianz* (*Variance*), *Standardabweichung* (*Standard deviation*), *Schiefe* (*Skewness*), und *Wölbung* (*Kurtosis*) wird im Anhang A erläutert. Der Median ist derjenige Wert, bei dem sich die nach Größe geordnete Menge der Ausgabewerte in zwei gleich stark besetzte Hälften teilt. Ebenso markieren die *Perzentile* diejenigen Werte, die diese Menge beim jeweils angegebenen Prozentsatz teilen. Die Perzentile können nur für durch 5 teilbare Prozentzahlen berechnet werden.

Der *Exceedance risk estimator* wird im Anhang B detailliert erklärt. Er schätzt mit der angegebenen Aussagesicherheit das Risiko ab, dass der angegebene Grenzwert

überschritten wird. Sofern es keine Überschreitungen gibt, wird ein fester, von der Zahl der Spiele und dem Vertrauenswert abhängiger Wert berechnet. Um aussagekräftigere Zeitverläufe zu erhalten, kann es deshalb sinnvoll sein, einen Schwellenwert einzutragen, der deutlich unter dem geforderten Grenzwert liegt.

Bei der Durchführung der analytischen Unsicherheitsanalyse werden im festgelegten Ausgabeverzeichnis Dateien mit dem Namen *[Basisname]_tUA* und der dateitypspezifischen Erweiterung (*.csv* oder *.dat*) erzeugt, welche die Ungewissheitsmaße für die ausgewählten Zeitpunkte enthalten. Falls die Maximum-Auswertung gewählt wurde, so werden zusätzlich die Ungewissheitsmaße für die Rechenlauf-Maxima in Tabellenform in die **.protocol*-Datei geschrieben.

5.4.2 Grafische Ungewissheitsanalyse

Der Button *Graphic UA* öffnet das in Abb. 5.6 dargestellte Dialogfenster. In der Auswahlliste stehen derzeit vier grafische Methoden zur Verfügung.

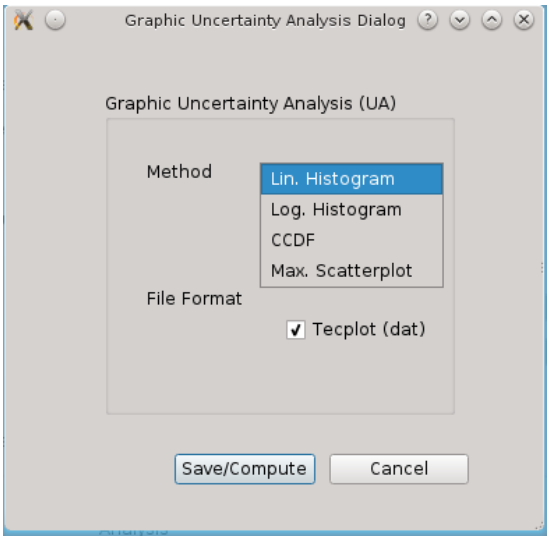


Abb. 5.6 Dialogfenster für die grafische Ungewissheitsanalyse

Mithilfe von Histogrammen kann die Verteilung der errechneten Modellausgabewerte sichtbar gemacht werden. Bei Verteilungen über mehrere Größenordnungen, wie sie z. B. für Dosisberechnungen typisch sind, sollten logarithmisch skalierte Histogramme verwendet werden. Die Klassenrasterung wird automatisch optimiert, der Benutzer hat hierauf keinen Einfluss. Alle ausgewählten Zeitpunkte werden zum Zweck der Vergleichbarkeit auf demselben Klassenraster ausgewertet. Falls Maximumauswertung

gewählt wurde, wird dafür jedoch ein eigenes Klassenraster verwendet. Die Berechnung der Histogramme ist in Anhang C ausführlich erläutert. Die Darstellungsdaten für Histogramme werden in den Dateien *[Basisname]_tHist* bzw. *[Basisname]_mHist* mit den typspezifischen Erweiterungen gespeichert.

Die komplementäre kumulierte Dichtefunktion (CCDF) gibt an, welcher Anteil aller Spiele bezüglich des Modellausgabewerts unterhalb des auf der *x*-Achse aufgetragenen Wertes bleibt. Derzeit werden nur die Rechenlaufmaxima, nicht aber die ausgewählten Zeitpunkte ausgewertet. Die Kurve wird auch als nicht-komplementäre Version (CDF) berechnet. Die Daten für beide Kurven werden in den Dateien *[Basisname]_mCCDF* mit den dateitypspezifischen Erweiterungen gespeichert.

Die Option „Maximum Scatterplot“ dient zum Erstellen eines Streudiagramms, bei dem alle Rechenlaufmaxima über dem Zeitpunkt ihres Auftretens dargestellt werden. In die Dateien *[Basisname]_mScat* mit den typspezifischen Erweiterungen werden dabei zusätzlich die Nummern aller gerechneten Radionuklide ausgegeben, und zwar in der Reihenfolge, die sich aus der Höhe ihrer Werte zum jeweiligen Maximalzeitpunkt ergibt. Mithilfe dieser Information lassen sich die Punkte des Diagramms nach den im Maximum jeweils dominierenden Radionukliden einfärben.

5.4.3 Analytische Sensitivitätsanalyse

Der Button „Analytic SA“ öffnet das in Abb. 5.7 dargestellte Dialogfenster. Im oberen Bereich können verschiedene Methoden der Sensitivitätsanalyse ausgewählt werden. Auf der linken Seite sind dabei korrelations- und regressionsbasierte Methoden aufgeführt, auf der rechten Seite nichtparametrische und varianzbasierte Methoden. Angeboten werden nur diejenigen Methoden, die mit dem verwendeten Ziehungsschema kompatibel sind.

Das Programm erzeugt bei der Ausführung Dateien mit Namen nach dem Muster *[Basisname]_t[Methode]* für die zeitabhängige Darstellung. Die Ergebnisse der Maximum-Auswertung einschließlich der Rangfolge der Variablen werden in Tabellenform in der **.protocol*-Datei abgelegt.

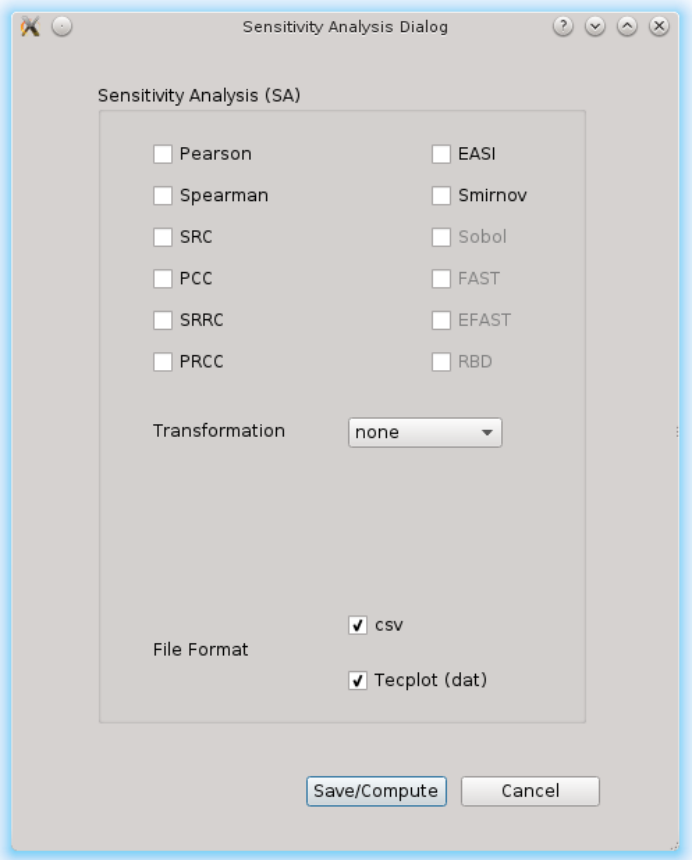


Abb. 5.7 Dialogfenster für die analytische Sensitivitätsanalyse

5.4.3.1 Auswerteverfahren

An dieser Stelle wird nicht im Detail auf die verschiedenen Methoden eingegangen, siehe dazu /SPI 16/. Die folgenden kurzen Ausführungen sollen dem Benutzer als Anwendungshilfe dienen.

Die Korrelationskoeffizienten nach Pearson, die standardisierten Regressionskoeffizienten (SRC) und die partiellen Korrelationskoeffizienten (PCC) sind verwandte Sensitivitätsmaße, die den linearen Einfluss der verschiedenen Variablen auf die Modellausgabegröße bewerten. Die Werte dieser Koeffizienten liegen zwischen -1 und 1, wobei +1 strenge lineare Abhängigkeit, 0 Unabhängigkeit und -1 strenge invers-lineare Abhängigkeit bedeutet. Diese Verfahren setzen eine gewisse Linearität des Modells voraus. Inwieweit diese gegeben ist, lässt sich an dem Bestimmtheitsmaß R^2 ablesen, das bei Berechnung von SRC mit ausgegeben wird. Als Faustregel kann gelten, dass R^2 größer als 0,5 sein sollte.

Es ist zu beachten, das SRC und PCC bei nichtkorrelierten Eingangsvariablen mathematisch äquivalent sind. Eventuelle Abweichungen bei den Ergebnissen sind dann auf Scheinkorrelationen zurückzuführen. Bei einer Analyse mit korrelierten Eingangsvariablen kann ein Vergleich zwischen beiden Maßen sinnvoll sein.

Für stark nichtlineare Modelle sind die genannten Verfahren prinzipiell weniger geeignet. Durch eine Rangtransformation, bei der jeder Variablen- oder Modellausgabewert durch seine Position in der Rangliste ersetzt wird, lassen sich monotone Zusammenhänge in lineare überführen. Dies wird bei den rangbasierten Versionen der oben genannten Koeffizienten (Rangkorrelationskoeffizienten nach Spearman, Standardisierte Rangregressionskoeffizienten SRRC und partielle Rangkorrelationskoeffizienten PRCC) automatisch durchgeführt. Dadurch verbessert sich im Allgemeinen die Modellbestimmtheit, ablesbar am Bestimmtheitsmaß R^2, für den Preis eines Verlustes an quantitativer Aussagekraft der Sensitivitätsmaße.

Der Smirnov-Test ist ein nichtparametrischer Test, bei dem die Ausgabewerte im Verhältnis 90:10 in zwei Gruppen mit niedrigen bzw. hohen Werten aufgeteilt werden. Mittels eines statistischen Tests werden dann die Verteilungen der den beiden Teilproben zugrunde liegenden Variablenwerte auf Übereinstimmung geprüft. Je größer die Abweichung ist, desto höher fällt der berechnete Wert aus, der somit als Sensitivitätsmaß dienen kann. Dieses Maß liegt zwischen 0 und 1. Da die Ergebnisse des Smirnov-Tests quantitativ nicht leicht zu interpretieren sind und die feste 90:10-Einteilung nicht immer problemgerecht ist, sollte dieses Maß nur mit Vorsicht zur Bewertung herangezogen werden.

Bei EASI, Sobol, FAST, EFAST und RBD handelt es sich um varianzbasierte Verfahren der Sensitivitätsanalyse. Diese Methoden berechnen prinzipiell dieselben Maße, nämlich die varianzbasierten Sensitivitätskoeffizienten erster oder höherer Ordnung, jedoch nach unterschiedlichen mathematischen Verfahren. Alle Koeffizienten liegen zwischen 0 (keine Abhängigkeit) und 1 (strenge Abhängigkeit). Die Koeffizienten erster Ordnung bewerten den Einfluss der einzelnen Variablen isoliert, während diejenigen höherer Ordnung Wechselwirkungen zwischen zwei oder mehr Variablen berücksichtigen. Die Koeffizienten der so genannten totalen Ordnung geben an, wie sensitiv das Modell gegenüber Änderungen einer Variablen im Zusammenspiel mit allen anderen ist. Während die Verfahren EASI , FAST und RBD nur die Koeffizienten erster Ordnung berechnen, liefert EFAST auch die Koeffizienten totaler Ordnung, und das Sobol-Verfahren kann prinzipiell alle Ordnungen bestimmen.

Sobol, FAST, EFAST und RBD arbeiten mit speziellen Abtastungen des Parameterraums, die bereits bei der Stichprobenziehung berücksichtigt werden müssen; diese Methoden der Sensitivitätsanalyse benötigen daher speziell auf sie zugeschnittene Ziehungsverfahren. Solche Ziehungsverfahren haben mehrere Nachteile. Sie sind für andere Auswertungen nicht oder nur eingeschränkt verwendbar, und die Stichproben sind im Allgemeinen weder teil- noch erweiterbar. Außerdem ergeben diese Verfahren zum Teil eine sehr inhomogene Abdeckung des Variablenraums, was zu wenig robusten Ergebnissen führen kann. Dagegen ist die EASI-Methode mit jedem Zufalls-, Quasi-Zufalls- oder geschichteten Ziehungsverfahren anwendbar.

5.4.3.2 Transformation

Typische Ergebnisse von Modellrechnung für Endlagersysteme ergeben weit verteilte Ausgabewerte, die sich über mehrere Größenordnungen erstrecken, wobei die allermeisten Werte sehr niedrig oder sogar null sind, die wenigen höheren aber gerade die interessanten Fälle darstellen. Die Gesamtvarianz der Ergebnisse wird dann stark durch solche Extremwerte dominiert.

Durch eine geeignete Transformation können die Daten auf eine sinnvollere, für Varianzauswertung besser geeignete Skala projiziert werden, wodurch die Aussagekraft und Robustheit der Sensitivitätsanalyse u. U. deutlich erhöht werden kann /SPI 16/. Über den Button *Transformation* können zwei verschiedene Transformationen ausgewählt werden, die dann vor der Auswertung auf die Modellausgabewerte angewendet werden:

logarithmisch: $y \mapsto \log_2\left(\frac{y}{a}+1\right)$

Potenz: $y \mapsto \left(\frac{y}{a}\right)^p$

Für diese Transformationen wird ein Schwellenwert a benötigt. Dieser sollte so gewählt werden, dass er den Übergang zwischen „kleinen“ und „großen“ Werten markiert, also eine Schwelle, oberhalb der die Werte für die Auswertung besonders relevant erscheinen. Die Transformationen bilden null auf null und kleine Werte auf Werte unter 1 ab, große Werte werden auf einen Wertebereich über 1 abgebildet. Im Gegensatz zu einer einfachen Logarithmierung verursachen diese Transformationen keine Übergewichtung

extrem kleiner Werte und lassen auch Nullwerte zu. Sinnvolle Exponenten für die Potenztransformation sind etwa 0,2 oder 0,3.

5.4.4 Grafische Sensitivitätsanalyse

Der Button *Graphic SA* öffnet das in Abb. 5.8 dargestellte Dialogfenster. Die Auswahlliste bietet vier grafische Verfahren an.

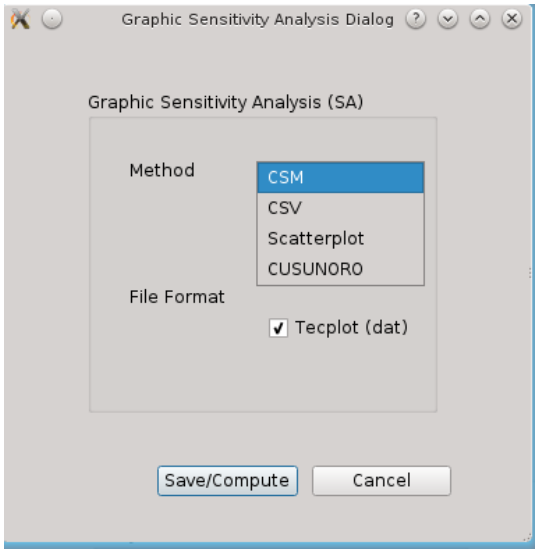


Abb. 5.8 Dialogfenster für die Grafische Sensitivitätsanalyse

Bei CSM („Contribution to the Sample Mean“) werden die Modellausgabewerte nach der Größe des untersuchten Variablenwertes sortiert. Aufgetragen wird dann der bis zu diesem Punkt kumulierte relative Beitrag zum Mittelwert der Ausgabewerte über dem Anteil an der Gesamtprobe. Sofern keine Abhängigkeit von der untersuchten Variable besteht, steigt dieser Beitrag gleichmäßig an, andernfalls verändert sich die Steigung. Das Maß für die Sensitivität ist also die Krümmung der Kurve. Die CSM-Kurven aller Variablen beginnen im Punkt (0,0) und enden im Punkt (1,1), dazwischen weichen sie mehr oder weniger stark von der Diagonalen ab. Stark gekrümmte Kurven weisen auf eine hohe Sensitivität hin.

Bei CSV („Contribution to the Sample Variance“) wird anstelle des kumulierten Beitrags zum Mittelwert derjenige zur Gesamtvarianz über dem Anteil an der Gesamtprobe aufgetragen. Die Aussage ist ähnlich wie bei CSM, die Kurven sind meist ausgeprägter gekrümmt, erscheinen aber aufgrund der Stichprobenstreuungen weniger glatt.

CUSUNORO („Cumulated Sum of Normalised Reordered Output“) /PLI 12/ stellt eine mathematisch optimierte Version des CSM-Plots dar, bei der gewisse Mängel der CSM-Darstellung vermieden werden. Alle Kurven beginnen bei (0,0) und enden bei (1,0). Eine auffällige Abweichung von der *x*-Achse ist ein Hinweis auf eine hohe Sensitivität des Modells gegenüber der betreffenden Variablen. Auch hier stellt die Krümmung der Kurven das eigentliche Sensitivitätsmaß dar.

CSM, CSV und CUSUNORO erlauben, anders als die numerischen Sensitivitätsmaße, eine Beurteilung der lokalen Sensitivität im Bereich bestimmter Werte der Variablen. Wenn eine Kurve einen mehr oder weniger scharfen Knick aufweist, liegt eine lokale Sensitivität bei demjenigen Wert vor, dessen Quantil auf der *x*-Achse abzulesen ist.

Als vierte Option zur grafischen Sensitivitätsanalyse wird das einfache Streudiagramm (Scatterplot) angeboten. Dabei wird für alle Spiele der Wert der Modellausgabegröße über dem Variablenwert dargestellt. Die Darstellung erlaubt eine direkte visuelle Beurteilung der Sensitivität.

Das Programm erzeugt bei der Auswertung Dateien mit Namen nach dem Muster *[Basisname]_t[Methode]* für die zeitabhängige Darstellung bzw. *[Basisname]_m[Methode]* für die Auswertung der Maxima, jeweils mit den typspezifischen Erweiterungen.

Literatur

/BEC 08/ Becker, D.-A.; Fein, E; Mönig; J.: Protocol for Assessing Parameter Uncertainty. PAMINA Milestone M-2.2.A.2. Via: Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, Braunschweig, 2008.

/HIR 99/ Hirsekorn, R.-P., Boese, B. und Buhmann, D.: *LOPOS*: Programm zur Berechnung der Schadstofffreisetzung aus netzwerkartigen Grubengebäuden, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH, GRS 157, Braunschweig, 1999.

/REI 11/ Anpassung des Programmpakets EMOS an moderne Softwareanforderungen, ADEMOS - Phase 1, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH, GRS-A-3623, BMWi-FKZ 02E10367, Braunschweig, 2011.

/REI 16/ Reiche, T.: *RepoTREND* – Das Programmpaket zur integrierten Langzeitsicherheitsanalyse von Endlagersystemen, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, GRS-411, Braunschweig 2016

/SPI 16/ Spießl, S., Becker, D.-A.: Investigation of Modern Methods of Probabilistic Sensitivity Analysis of Final Repository Performance Assessment Models (MOSEL), Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, GRS-412, Braunschweig 2016 (in Vorbereitung)

/CER 16/ Cerrani, I.: *SimLab4* Dll version user guide (ver 1.1.0). Preliminary version (unpublished), 2016

/SIM 04/ *SimLab 2.2* REFERENCE MANUAL, Joint Research Centre, Ispra, 2004

/PLI 12/ Plischke, E.: An adaptive correlation ratio method using the cumulative sum of the reordered output. Reliability Engineering and System Safety 107 (2012) 149–156

/WWW-R/ The Comprehensive R Archive Network. https://cran.r-project.org/

Abbildungsverzeichnis

Anhang A Mittelwert, Standardabweichung, Schiefe und Wölbung

Statistische Verteilungen können durch ihren *Erwartungswert* $\mu = E(X)$ und die *zentralen Momente* μ_k charakterisiert werden:

$$\mu_k = E((X-\mu)^k) \qquad (1)$$

Wegen $E(X) = \mu$ ist das 1. zentrale Moment immer null. Das zweite zentrale Moment ist die *Varianz*:

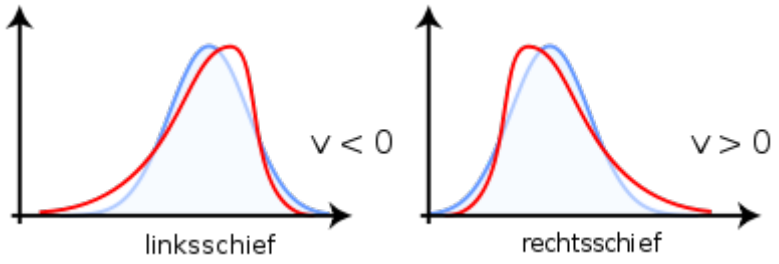
$$v = \mu_2 = E((X-\mu)^2) \qquad (2)$$

$$\sigma = \sqrt{\mu_2} \qquad (3)$$

Die Wurzel aus der Varianz (die nicht negativ sein kann) ist die Standardabweichung σ und charakterisiert die Breite der Verteilung. Das dritte zentrale Moment stellt ein Maß für die Abweichung der Verteilung von einer symmetrischen Gestalt dar und wird in einer auf σ normierten Form als *Schiefe* ν bezeichnet:

$$\nu = \frac{\mu_3}{\sigma} = E\left(\frac{(X-\mu)^3}{\sigma}\right) \qquad (4)$$

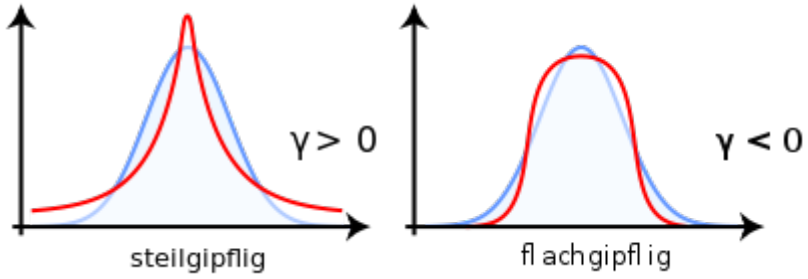
Eine Verteilung mit $\nu < 0$ heißt *linksschief*, mit $\nu > 0$ *rechtsschief*, siehe folgende Abbildung (aus Wikipedia):



Das vierte zentrale Moment beschreibt, wie stark die Verteilungskurve gewölbt ist, und heißt in der auf σ normierten Form dementsprechend *Wölbung* oder *Kurtosis*:

$$w = \frac{\mu_4}{\sigma} = E\left(\frac{(X-\mu)^4}{\sigma}\right) \qquad (5)$$

Da die Wölbung der Normalverteilung 3 beträgt, ist es sinnvoll, das Maß $\gamma = w - 3$ einzuführen. Dies wird als *Exzess* bezeichnet. Damit lässt sich zwischen *steilgipfligen* ($\gamma > 0$) und *flachgipfligen* ($\gamma < 0$) Verteilungen unterscheiden, siehe folgende Abbildung (aus Wikipedia):



Die Bezeichnung *Kurtosis* wird allerdings nicht einheitlich verwendet; häufig ist damit der Exzess gemeint, so z. B. bei EXCEL®. Auch *R* und somit *SimLab4* sowie schließlich *RepoSUN* liefern den unter der Bezeichnung Kurtosis den Exzess zurück.

Bei probabilistischen Unsicherheitsanalysen ist die exakte Verteilungsfunktion nicht bekannt. Stattdessen liegt eine mehr oder weniger umfangreiche Stichprobe von Rechenergebnissen vor, die mit nach Expertenvorgaben verteilten Eingangsparametern bestimmt wurden. Der Erwartungswert und die Momente der tatsächlichen Verteilung können daher nicht berechnet, sondern nur anhand der vorhandenen Stichprobe geschätzt werden. Der Anwender erwartet solche abschätzenden Aussagen über die unbekannte tatsächliche Verteilung und nicht etwa exakte Aussagen über die konkrete, aber zufallsbestimmte Stichprobe. Deshalb kommen für die Berechnung der Momente aus den Stichprobendaten modifizierte Formeln zum Einsatz. Da aber neben einer gewissen Begriffsverwirrung häufig Unklarheit über die anzuwendenden Formeln besteht, werden diese hier zusammengestellt. Mathematische Begründungen werden dabei nicht gegeben.

Es wird angenommen, dass eine Stichprobe von n Rechenergebnissen $x_1, \ldots, x_n$ vorliegt, die die tatsächliche Verteilung repräsentieren.

Als Schätzer für den Erwartungswert μ dient der arithmetische *Mittelwert* $\bar{x}$:

$$\langle \mu \rangle = \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \qquad (6)$$

Das ist bereits der ideale Schätzwert, eine Modifikation gibt es daher nicht. Für die Berechnung eines Schätzers für die Standardabweichung als Wurzel aus der Varianz sind jedoch zwei Formeln üblich:

$$\langle \sigma \rangle = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2} \qquad (7)$$

bzw.

$$\langle \sigma \rangle = s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2} \qquad (8)$$

Die Formel (7) ist anzuwenden, wenn der tatsächliche Erwartungswert μ der Verteilung bekannt ist. Das gilt z. B. dann, wenn die Werte $x_1, ..., x_n$ vollständig sind, also keiner Stichprobe, sondern einer Gesamterhebung entsprechen. In diesem Fall ist $\mu = \bar{x}$ und die linke Formel gilt exakt, d. h. $\sigma = s' = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2}$. Berechnet man diesen Wert jedoch für eine echte Stichprobe, dann spricht man von der *unkorrigierten Stichproben-Standardabweichung*. Diese ist die Standardabweichung der Stichprobe selbst als vollständiger Grundgesamtheit, stellt aber keinen optimalen Schätzer für die tatsächliche Standardabweichung einer durch diese repräsentierten unbekannten Verteilung dar. Da der wahre Erwartungswert einer unbekannten Verteilung niemals bekannt ist, ist in diesem Fall Formel (8) anzuwenden, die die *korrigierte Stichproben-Standardabweichung* s liefert. Es gilt

$$s = \sqrt{\frac{n}{n-1}} s' \qquad (9)$$

Die korrigierte ist also stets etwas größer als die unkorrigierte Standardabweichung. Für die Schiefe und die Wölbung gelten ähnliche Überlegungen. Es existieren jeweils unkorrigierte und korrigierte Formeln, von denen die ersteren exakte Werte für die Stichprobe als Grundgesamtheit liefern, die letzteren jedoch optimale Schätzer für die

Charakteristika der unbekannten Verteilung liefern. Die Formeln (aus Wikipedia) sind in der Tab. A.1 zusammengefasst.

Da bei probabilistischen Analysen immer eine unbekannte wahre Verteilung charakterisiert werden soll, werden grundsätzlich die korrigierten Formeln angewandt.

Tab. A.1 Unkorrigierte und korrigierte Formeln für statistische Charakteristika

Charakteristikum	Unkorrigierte Formel	Korrigierte Formel	Umrechnung
Standard-abweichung	$s' = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$	$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$	$s = \sqrt{\frac{n}{n-1}}s'$
Schiefe	$v' = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s'}\right)^3$	$v = \frac{n}{(n-1)(n-2)}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s}\right)^3$	$v = \frac{\sqrt{n(n-1)}}{n-2}v'$
Exzess (bezeichnet als Kurto-sis)	$w' = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s'}\right)^4 - 3$	$w = \frac{n(n+1)}{(n-1)(n-2)(n-3)}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s}\right)^4 - \frac{3(n-1)^2}{(n-2)(n-3)}$	$w = \frac{(n-1)^2}{(n-2)(n-3)}\left(\frac{n+1}{n-1}(w'+3)\right) - 3$

Anmerkung: MS Excel® bietet für die Standardabweichung beide Formeln an, wobei die korrigierte Standardabweichung in der deutschen Version STABW.S, die unkorrigierte STABW.N heißt. Für Schiefe und Wölbung (SCHIEFE bzw. KURT) stehen nur die korrigierten Formeln zur Verfügung. Die Funktion KURT liefert tatsächlich nicht die Wölbung oder Kurtosis, sondern den Exzess. *SimLab4* greift bei der Berechnung auf *R*-Routinen zurück. Dabei wird für die Standardabweichung („StdDev“) die korrigierte Formel verwendet, für Schiefe („Skewness“) und Wölbung („Kurtosis“) aber die unkorrigierten Formeln. Anstelle der Kurtsosis liefert *SimLab4* ebenfalls den Exzess

Anhang B Berechnung eines Schätzers für das Überschreitensrisiko (exceedance risk estimator)

Mithilfe der Binomialverteilung kann berechnet werden, wie viele zufällig gezogene Spiele mit welcher Maximalzahl von Grenzwertverletzungen erforderlich sind, um ein bestimmtes Kriterium – typischerweise 95/95 oder 99/95 – zu erfüllen. Dies ist so zu lesen, dass mit 95-%iger Aussagesicherheit mindestens 95 % (99 %) aller möglichen Rechnungen unter dem Grenzwert bleiben. So ergibt sich z. B. für das 95/95-Kriterium, dass dieses bei 59 zufällig ausgewählten Rechenläufen ohne eine einzige Überschreitung erfüllt ist.

Für eine stetige Auswertung ist ein solches diskretes Kriterium wenig geeignet. Bei fester Vorgabe kann seine Einhaltung zwar überprüft werden, dies liefert aber nur eine Ja/Nein- Aussage, und zwar fast immer Ja, weil das System so konstruiert ist, dass das Kriterium möglichst immer eingehalten wird. Daraus ist also kaum Information abzuleiten. Stattdessen erscheint es informativer, ein Überschreitensrisiko zu berechnen, das dann auch z. B. in seinem Zeitverlauf dargestellt werden kann. Dieses wird im Folgenden definiert und erläutert.

Die Verteilung der Ausgangswerte des langzeitsicherheitsanalytischen Rechenmodells ist unbekannt. Dies gilt somit auch für das Risiko, dass ein bestimmter Grenzwert überschritten wird. Dieses tatsächliche Überschreitensrisiko sei mit r bezeichnet. Gesucht wird nun ein Schätzer q für r in dem Sinne, dass er r möglichst, d. h. mit einer hohen Wahrscheinlichkeit a (Aussagesicherheit), nicht unterschätzen soll.

Die Binomialverteilung liefert eine Aussage über ein Zufallsexperiment mit zwei möglichen Ergebnissen („Erfolg“ und „Misserfolg“) mit bekannter Erfolgswahrscheinlichkeit r. Wenn das Experiment n-mal ausgeführt wird, dann ist

$$b_{n,r}(k) = \binom{n}{k} r^k (1-r)^{n-k} \qquad (10)$$

die Wahrscheinlichkeit, dass dabei *genau* k-mal das Ergebnis „Erfolg“ auftritt. Setzt man (rein formal) „Erfolg“ mit „Grenzwertüberschreitung“ gleich, so müsste man zur direkten Anwendung dieser Formel die Überschreitenswahrscheinlichkeit r kennen. Die Formel liefert dann die Wahrscheinlichkeit, bei n zufälligen Spielen genau k Überschreitensfälle zu erhalten. Tatsächlich ist die Situation aber umgekehrt: r ist nicht be-

kannt und soll anhand der beobachteten Anzahl k von Überschreitungen abgeschätzt werden. Dafür wird die kumulierte Binomialverteilung verwendet:

$$B_{n,r}(k) = \sum_{j=0}^{k} b_{n,r}(j) = \sum_{j=0}^{k} \binom{n}{j} r^j (1-r)^{n-j} \quad . \qquad (11)$$

Sie beschreibt die Wahrscheinlichkeit, dass bei n Zufallsexperimenten mit der Erfolgswahrscheinlichkeit r *höchstens* k-mal das Ergebnis „Erfolg“ auftritt. Angenommen, bei einer statistischen Simulation mit n Spielen werden k Grenzwertüberschreitungen festgestellt, wobei die tatsächliche Wahrscheinlichkeit r, bei einem einzelnen Spiel den Grenzwert zu überschreiten, nicht bekannt ist. Diese soll wie oben erläutert mit der Aussagesicherheit a abgeschätzt werden. Dazu ist der Wert q so zu bestimmen, dass die Beziehung

$$B_{n,q}(k) = 1 - a \qquad (12)$$

erfüllt ist. Da die Formel nicht geschlossen auflösbar ist, muss q mit einem Newton-Verfahren approximiert werden. Dann stellt q einen Schätzer für r dar, der die oben genannten Anforderungen erfüllt. Das kann folgendermaßen verstanden werden:

Wäre das Überschreitensrisiko r tatsächlich gleich q, dann würde die Formel (12) die Wahrscheinlichkeit angeben, höchstens k Überschreitungen zu finden. Da a nahe bei 1 liegt (z. B. 0,95), ist diese Wahrscheinlichkeit gering, d. h. es würde höchstwahrscheinlich mehr als k Überschreitungen geben. Da aber nur k Überschreitungen festgestellt wurden, ist es wahrscheinlich, dass r durch q überschätzt wird.

Sei $p = B_{n,r}(k)$ die unbekannte tatsächliche Wahrscheinlichkeit für das festgestellte oder ein besseres Ergebnis (also k oder weniger Überschreitungen). Aus der *Annahme* $r > q$ folgt dann $p < 1 - a$. Im logischen Umkehrschluss folgt aus $p \geq 1 - a$ die Beziehung $r \leq q$. Die Wahrscheinlichkeit, dass p größer oder gleich $1 - a$ ist, beträgt aber gerade a. Das bedeutet, dass q mit der Wahrscheinlichkeit a eine obere Grenze für r darstellt, was gefordert war.

Der Schätzer q leitet sich rein mathematisch aus den statistischen Ergebnissen ab, hat aber mit dem Rechenmodell nichts zu tun und muss somit keineswegs der tatsächlichen Überschreitenswahrscheinlichkeit r nahe kommen. Sollten z. B. modellbedingt

überhaupt keine Grenzwertüberschreitungen möglich sein, dann kann die Überschreitenswahrscheinlichkeit sogar exakt null sein, was für den Schätzer jedoch nicht gilt.

Da weniger als 0 Überschreitungen nicht möglich sind, folgt aus der Art der Berechnung, dass ein Minimalwert $q_{\min}$ für den Schätzer existiert. Dieser errechnet sich durch Auflösen der Gleichung

$$B_{n,q_{\min}}(0) = 1 - a \qquad (13)$$

nach $q_{\min}$. So ergibt sich z. B. für $a = 0{,}95$ und $n = 3000$ der Minimalwert $q_{\min} = 0{,}000998$. Kommen also keine Überschreitungen vor, dann erhält man immer diesen Wert, egal, wie weit die Modellergebnisse vom Grenzwert tatsächlich entfernt sind.

Anhang C Erstellung von Häufigkeits-Histogrammen

Zur Darstellung der Häufigkeitsverteilung von statistischen Rechenergebnissen werden Histogramme verwendet. Dazu wird der Wertebereich in eine Anzahl von Klassen eingeteilt. Die absolute Belegungszahl jeder Klasse wird dann in Form einer Säule über dem Teil-Wertebereich dargestellt.

Histogramme können für einen festen Zeitpunkt oder für das Maximum über alle Zeitpunkte erstellt werden. Für die praktische Umsetzung waren einige Fragen zu klären und technische Probleme zu lösen. Dies wird im Folgenden dargestellt.

Zunächst sollen grundsätzlich alle dargestellten Klassen die gleiche Breite haben. Des Weiteren ist über die Anzahl der Klassen zu entscheiden. Bei zu wenigen Klassen verliert das Diagramm seine Aussagekraft. Bei zu vielen Klassen werden diese jedoch so schmal, dass sie nur noch mit wenigen Werten besetzt sind. Das führt dazu, dass das Diagramm zunehmend zufallsbestimmt wird und keinen klaren Verlauf mehr erkennen lässt. Im Extremfall gäbe es fast nur noch Klassen mit einem oder keinem Element. Als sinnvoller Wert für die Anzahl der k Klassen erscheint die (ganzzahlig gerundete) Wurzel aus der Anzahl der Werte:

$$k = \left\lfloor \sqrt{N} + 0.5 \right\rfloor \qquad (14)$$

Im für Endlagersimulationen typischen Bereich zwischen 100 und 10000 Spielen erhält man damit zwischen 10 und 100 Klassen, die im Mittel mit jeweils 10 bis 100 Werten belegt sind. Diese Formel ist im Programmcode von *post-statist* fest implementiert.

Als Nächstes ist die Klassenbreite zu bestimmen. Dabei ist grundsätzlich zu unterscheiden, ob das Histogramm auf einer linearen oder einer logarithmischen Skala dargestellt werden soll. Auf der linearen Skala kann der gesamte darzustellende Wertebereich einfach durch Differenzbildung zwischen dem größten und dem kleinsten auftretenden Wert ermittelt werden. Die Klassenbreite b ergibt sich dann durch Division durch die Klassenanzahl als

$$b = \frac{y_{\max} - y_{\min}}{k}. \qquad (15)$$

Auf einer logarithmischen Skala müssen die Klassen zu größeren Werten hin auch breiter werden, damit sie in der Darstellung gleich breit erscheinen. Anstelle einer konstanten Klassenbreite ist ein Faktor β festzulegen, der den oberen mit dem unteren Randwert jeder Klasse verknüpft. Die Bestimmung des gesamten darzustellenden Bereichs ist problematisch, weil Nullen und in Sonderfällen sogar negative Werte auftreten können. Solche Werte sind nicht darstellbar. Darüber hinaus kann ein numerisches Modell aber auch zufallsbestimmte, sehr kleine positive Werte liefern, die faktisch als Nullwerte anzusehen sind. Bei einheitlich-formaler Behandlung auf der logarithmischen Skala würden solche Werte ggf. zu stark linkslastigen Skalen mit zahlreichen unbelegten Klassen führen. Es erscheint sinnvoll, sehr kleine sowie negative Werte als Nullwerte anzusehen und ebenso wie diese zu behandeln. Nullen würden im Histogramm „unendlich weit links“ liegen und sich dort über einen „unendlich breiten“ Bereich erstrecken. Da eine solche „Nullklasse“ aber nur mit endlich vielen Werten belegt ist, hat sie die Höhe null. Das bedeutet, dass diese Werte im Histogramm überhaupt nicht darzustellen sind. Deshalb wird für logarithmische Histogramme das folgende Verfahren verwendet:

Zunächst wird der höchste auftretende Einzelwert $y_{\max}$ ermittelt (es wird davon ausgegangen, dass allenfalls ausnahmsweise negative Werte vorkommen, $y_{\max}$ also positiv ist, andernfalls sollten alle Werte invertiert werden). Dieser bildet auf jeden Fall den oberen Rand des Darstellungsintervalls. Der untere Rand ist durch den kleinsten darstellungswürdigen Wert bestimmt. Die Darstellungswürdigkeit ist allerdings schwer objektiv zu erfassen. Dafür wird folgendes Kriterium verwendet:

Wenn in n benachbarten Klassen zusammen mindestens n Werte liegen, dann ist davon auszugehen, dass diese signifikant und darstellungswürdig sind.

Die Zahl n wird dabei als ganzzahlig gerundeter Zehnerlogarithmus der Gesamtzahl der Werte bestimmt:

$$n = |\log N + 0.5| \qquad (16)$$

Das bedeutet $n = 1$ für 4 bis 31 Werte, $n = 2$ für 32 bis 316 Werte, $n = 3$ für 317 bis 3162 Werte usw. Für weniger als 4 Werte ist ein Histogramm sinnlos.

Zur Überprüfung des Kriteriums werden zuerst die n kleinsten positiven Werte $y_1 \leq y_2 \leq \ldots \leq y_n$ ermittelt. Dann wird überprüft, ob

$$y_n < \log y_1 + n\frac{\log y_{\max} - \log y_1}{k} \qquad (17)$$

gilt. Wenn ja, dann ist das Kriterium erfüllt, und y_1 wird als kleinster darstellungswürdiger Wert angesehen. Andernfalls wird y_1 verworfen und durch y_2 ersetzt; y_2 wird durch y_3 und so weiter und y_n durch den nächstgrößeren Wert ersetzt. Damit wird das Kriterium erneut überprüft. Dies wird ggf. bis zum Erfolg wiederholt. Der Faktor β ist dann

$$\beta = \sqrt[k]{\frac{y_{\max}}{y_1}} = 10^{\frac{\log y_{\max} - \log y_1}{k}} \qquad (18)$$

mit dem zuletzt gefundenen Wert für y_1. Alle Werte unterhalb dieser Untergrenze bleiben im Histogramm unberücksichtigt. Die Anzahl dieser verworfenen Werte sollte aber als Information mit ausgegeben werden.

Die einzelnen Klassenintervalle müssen einseitig offen sein, damit sie sich nicht überlappen. Das führt jedoch zu einem Problem mit den Randwerten. Bei linksoffenen Intervallen wird der Minimalwert nicht mehr in der untersten Klasse erfasst, bei rechtsoffenen Intervallen liegt der Maximalwert außerhalb der obersten Klasse. Aus Gründen der Anschaulichkeit erscheint es wenig sinnvoll, die Darstellungsgrenzen über die tatsächlichen Minimal- bzw. Maximalwerte hinaus zu erweitern, denn das würde dem Betrachter einen falschen Eindruck von diesen Grenzen geben. Deshalb wird zwar prinzipiell mit rechtsoffenen Intervallen gearbeitet, das oberste Intervall wird aber ausnahmsweise als beidseitig geschlossen angenommen, sodass der Maximalwert in dieses Intervall fällt.

Bis hierher wurde die Darstellung eines Einzelhistogramms beschrieben. Werden zum selben Rechenlauf mehrere Zeitpunkte für die Darstellung ausgewählt, so ist es nicht besonders sinnvoll, die Histogrammdarstellungen unabhängig voneinander zu berechnen, denn das würde eine Bewertung der Zeitentwicklung erschweren oder sogar unmöglich machen. Es sollen deshalb für alle Zeitpunkte dieselben Klassengrenzen verwendet werden. Nach dem oben beschriebenen Verfahren können sich aber sehr verschiedene Klassenbreiten und Relevanzschwellen ergeben. Deshalb wird folgendermaßen vorgegangen:

Zunächst wird für jeden einzelnen Zeitpunkt nach dem beschriebenen Verfahren ohne Berücksichtigung der übrigen Zeitpunkte eine „optimale“ Klasseneinteilung berechnet. Als gemeinsame obere Darstellungsgrenze wird dann das Gesamtmaximum verwendet. Die gemeinsame untere Darstellungsgrenze ergibt sich je nach Histogrammtyp als das Gesamtminimum bzw. die kleinste berechnete Relevanzschwelle. Die gemeinsame Klassenbreite ist so zu bestimmen, dass sie möglichst wenig von den berechneten Optimalwerten abweicht. Dafür wird der Median aller dieser Werte verwendet; die sich daraus errechnende Klassenbreite wird dann so korrigiert, dass sie zu einer ganzen Gesamtanzahl von Klassen führt. Diese stimmt zwar nicht mehr mit der Wurzel aus der Zahl der Werte überein – tatsächlich kann sie bei weit auseinander liegenden Verteilungen sogar erheblich höher werden – aber jedes einzelne Histogramm erstreckt sich zumindest ungefähr über die Optimalanzahl von Klassen. Bei logarithmischen Histogrammen ist es auch möglich, dass das oben genannte Relevanzkriterium bezüglich der gemeinsamen Klasseneinteilung bei der kleinsten dargestellten Klasse nicht mehr erfüllt ist. Dies hat aber keine störenden Auswirkungen.

Anhang D Aufbau der Dateien **.svs* und **.sdo*

Die Vorspanndatei *.svs enthält die Informationen über den statistischen Rechenlauf, die für die Auswertung benötigt werden. Sie muss dem folgenden Schema entsprechen:

[Kommentarzeile, derzeit mit dem Rechenlaufnamen]
[weitere Kommentarzeile, derzeit mit Datum und Zeit]
%%UNITS
1
[Einheit]
%%ORT
1
[Bezeichnung der Modellausgabegröße]
%%NUKLID
[Gesamtzahl der Nuklide und Nuklidsummen]
[Auflistung aller Nuklide und Nuklidsummen. Je Zeile 8 Einträge à 10 Zeichen:
2-3 Leerzeichen, Elementname, 1-3 Leerzeichen, Massenzahl, 0-1 Leerzeichen,
relative Tochteradresse (Nuklide) bzw. Zahl der Summanden (Nuklidsummen)]
%%DIVERS
4
[Zahl der Nuklide, Zahl der Nuklidsummen, Zahl der Zeitpunkte, Zahl der Spiele]
%%SAMPLE
[vollständiger Pfad der **.sam*-Datei]
%%TIME
[Auflistung aller Zeitpunkte. Je Zeile 8 Einträge à 10 Zeichen]
%%RUNS
[Auflistung aller gerechneten Spielnummern. Ein Eintrag je Zeile]

Die Statistikdaten-Datei **.sdo* enthält alle Werte für eine Modellausgabegröße, die während des statistischen Rechenlaufs gesammelt wurden. Sie muss immer namensgleich im selben Verzeichnis wie die zugehörige Vorspanndatei stehen und mit dieser kompatibel sein. Sie ist folgendermaßen aufgebaut:

[Spiel-Nr, 5 Zeichen], [Zeitpunkt, 15 Zeichen]
[Auflistung der Ausgabedaten für alle Nuklide und Nuklidsummen. 8 Einträge je Zeile, 10 Zeichen je Eintrag]

[Wiederholung für alle Zeitpunkte]

[Wiederholung für alle Spiele]

Anhang E Programmkomponenten und Arbeitsumgebung

Um *RepoSTAR* nutzen zu können, müssen die erforderlichen Programmkomponenten korrekt installiert sein. Dies wird im Folgenden erläutert.

Die Datenversorgung und der Jobstart über die Benutzerschnittstelle *XENIA* finden interaktiv auf dem Arbeitsplatzrechner statt. Die eigentliche Rechnung erfolgt dann im Allgemeinen auf einem entfernten Batch-System. Die Auswertung mit *RepoSUN* geschieht interaktiv auf einem Linux-Rechner, also entweder dem Arbeitsplatzrechner oder mittels Fernzugriff. Somit sind bis zu drei Systeme zu unterscheiden, auf denen jeweils unterschiedliche Aufgaben ausgeführt werden und dementsprechend unterschiedliche Softwarekomponenten verfügbar sein müssen

Auf dem System, das für Datenversorgung und Jobstart verwendet wird, muss die Benutzerschnittstelle *XENIA* zur Verfügung stehen. Dazu gehört auch der Zugriff auf deren Datenbank. *XENIA* kann unter den beiden gängigen Betriebssystemen MS-Windows oder Linux ausgeführt werden, der Benutzer hat diesbezüglich freie Wahl. Neben den benötigten *RepoTREND*-Rechenmodulen sind für *RepoSTAR* zusätzlich die Rechenmodule *pre-statist* (falls eine Stichprobenziehung erfolgen soll) und *statist-control* erforderlich.

Falls eine Stichprobenziehung durchgeführt werden soll, also das Programm *pre-statist* zum Einsatz kommt, muss auf dem Rechensystem die dynamische Bibliothek *SimLab4* erreichbar sein. Diese Bibliothek stellt diverse Funktionalitäten zur Stichprobenziehung sowie zur Unsicherheits- und Sensitivitätsanalyse bereit und bindet dazu ihrerseits Skriptroutinen ein, die in der Statistik-Programmierumgebung *R* verfasst sind, weshalb auch *R* verfügbar sein muss. *R* muss dafür seinerseits durch Installation von Paketen erweitert werden. *SimLab4* greift auf die *R*-Pakete *lhs, RInside, sensitivity, pspearman, randtoolbox, e1071, mvtnorm, mc2d, rngWELL* und *stats* zu. Es ist sicherzustellen, dass diese korrekt installiert sind. Weiterhin bringt *SimLab4* das spezielle Paket *SimLab4R* mit, welches ebenfalls für *R* zu installieren ist.

Die Auswertung einer statistischen Rechnung geschieht unabhängig von der Rechnung selbst interaktiv mit dem Programm *RepoSUN*. Dieses steht derzeit nur für das Betriebssystem Linux zur Verfügung und muss deshalb auf einem Linux-Rechner ausgeführt werden. Sofern der verwendete Arbeitsplatzrechner keine eigene Linux-Umgebung besitzt, muss mittels eines X-Servers eine Remote-Verbindung zu einem

Linux-Rechner aufgebaut werden. Auf diesem muss die ausführbare Programmdatei *sensanalysis* für die grafische Benutzeroberfläche zur Verfügung stehen. Die *Repo-SUN*-Oberfläche bietet eine Auswahl von Linux-Zielrechnern an. Auf dem Zielrechner wird dann die eigentliche Analyse mit dem Programm *post-statist* ausgeführt, welches dort verfügbar sein muss. Da das Programm *post-statist* ebenfalls auf die dynamische Bibliothek *SimLab4* zugreift, müssen die gleichen Voraussetzungen wie für *pre-statist* hinsichtlich der Statistik-Programmierumgebung *R und deren Pakete* auf dem Zielrechner erfüllt sein wie im vorangegangenen Absatz bereits beschrieben.

Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH

Schwertnergasse 1
50667 Köln
Telefon +49 221 2068-0
Telefax +49 221 2068-888

Forschungszentrum
Boltzmannstraße 14
85748 Garching b. München
Telefon +49 89 32004-0
Telefax +49 89 32004-300

Kurfürstendamm 200
10719 Berlin
Telefon +49 30 88589-0
Telefax +49 30 88589-111

Theodor-Heuss-Straße 4
38122 Braunschweig
Telefon +49 531 8012-0
Telefax +49 531 8012-200

www.grs.de

ISBN 978-3-944161-93-8