

## Gebrauchsanleitung des SonArms GUI mit MultiSonArms 23-Feb-22.doc

10. Feb. 2021	Erstversion
09. Apr. 2021	Erweiterung: MultiSonArmsR.
23. Feb. 2022	Diverse kleine Verbesserungen des aktuellen GUI V4.4.0, ohne Funktionsänderungen. MultiSonArms.exe und MultiSonArmsR.exe mit neuer Versionsnummer V1.4.0. Keine Funktionsänderungen, basiert auf SonX 6.4.0.

### Inhalt

1.	Überblick .....	1
2.	Bedarf an Hardware .....	1
3.	Installation/Update .....	2
4.	Aufteilung in Subprojekte .....	2
5.	Start von MultiSonArms unter dem SonArms GUI .....	2
6.	Resultate und Verzeichnisse .....	4
7.	Abbrechen der Berechnungen .....	5
8.	Resultatunterschiede sind kein Fehler .....	6
9.	Referenzdokumente .....	6

## 1. Überblick

SonArms ist das vom BAFU bereitgestellte Programm zur Berechnung der Ausbreitung von Waffenlärm. Die Kenntnis von SonArms muss Ihnen hier zugemutet werden. (SonArms wird sehr gut in der Dokumentation von Jean Marc Wunderli (EMPA) erklärt. Diese liegt dem öffentlich verfügbaren BAFU Paket bei [1].)

Die Motivation für den Zusatz MultiSonArms ist die angestrebte Beschleunigung der SonArms Berechnungen. SonArms kennt den Rastermodus zur Berechnung von Lärmkarten. Dieser braucht bei grossen Gebieten wegen den zahlreichen Empfangspunkten sehr viel Rechenzeit, manchmal Tage oder gar Wochen. Deswegen ist MultiSonArms zugeschnitten für Rasterberechnungen, also Lärmkarten, nicht für Punktberechnungen, die sehr viel schneller ablaufen.

Schon seit geraumer Zeit enthalten aktuelle Prozessoren mehrere bis viele CPUs (Central Processing Units). MultiSonArms nutzt mehrere CPUs gleichzeitig und behebt damit den stossenden Umstand, dass mehrere CPUs vom heutigen Rechenkern nicht richtig ausgenutzt werden. MultiSonArms in Kombination mit dem Rechenkern beschleunigte die Rechenzeit in einem Beispiel mit 8 CPUs um den Faktor 5.

MultiSonArms führt eine ursprüngliche Projektdatei so aus, wie wenn es selber ein SonArms\_Kernel wäre und startet dafür den Rechenkern mehrfach. Es teilt dafür ein .wlp Projekt auf in mehrere Subprojekte, lässt sie mehrfach unter dem Rechenkern laufen und setzt am Ende die .wlm Resultate der Subprojekte wieder zusammen. Für die Resultate spielt es grundsätzlich keine Rolle, ob sie wie bisher als Einzelprozess berechnet worden sind oder ob sie neu parallel berechnet worden sind. Es kann aber in einigen Fällen kleine Unterschiede in den Resultatwerten geben. Die Gründe dafür erfahren Sie weiter unten (8).

Der SonArms Rechenkern und auch MultiSonArms werden normalerweise aus dem SonArms GUI (Graphical User Interface) heraus gestartet (ab GUIsonArms.exe V4.2.0 vom 19. Mrz. 2021). Bei Bedarf (z.B. für Berechnungsserien) kann das ohne GUI direkt aus einem selbst zu erstellenden Batch Programm heraus erfolgen oder Sie können einen misslungenen Subprozess reparieren. Das wird hier nicht gezeigt, eine Anleitung [2] dazu ist verfügbar.

## 2. Bedarf an Hardware

Um die hier behandelte Beschleunigungsmöglichkeit nutzen zu können, braucht es die nötige Hardware, die heutzutage allerdings fast überall verfügbar ist:

- 1) Mindestens einige CPUs im Prozessor. Der Windows Task Manager zeigt im Fenster „Leistung“ die Zahl der verfügbaren CPUs als eine Reihe von Fensterchen mit der individuellen CPU-Auslastung.

- 2) Pro CPU (bzw. Prozess) sollten etwa 2 GB RAM zur Verfügung stehen. Steht zu wenig Speicher zur Verfügung, wird Speicher auf den Disk ausgelagert, was die Leistung spürbar bremst. Dann ist es Zeit für einen Speicherausbau. Der Windows Task Manager zeigt die Speicher-Auslastung.
- 3) Ein Solid State Disk (SSD) anstelle einer Festplatte (HDD) ist möglicherweise hilfreich, aber wir vermuten, dass dies nicht viel ausmacht. Bei einer grossen Zahl CPUs könnte das ins Gewicht fallen.

### 3. Installation/Update

Eine vollständige Neuinstallation erfolgt durch simple Kopie aller Dateien und Verzeichnisse aus dem Gesamtpaket. Für ein Update Ihrer Installation reicht es, wenn Sie die beiden Programme ins GUI Verzeichnis, das Sie früher einmal angelegt haben, kopieren:

MultiSonArms.exe	(aktuell Version V1.4.0)
MultiSonArmsR.exe	(aktuell Version V1.4.0, für Reparatur von Resultaten, erklärt in [2])
GUIsonArms.exe	(aktuell Version V4.4.0)

Sie überschreiben damit die allfällig vorhandenen früheren Versionen.  
Das GUI Verzeichnis heisst typischerweise

....\BinGUI\

### 4. Aufteilung in Subprojekte

Bevor es an den Start geht, sollten sie wissen, dass MultiSonArms im Hintergrund Ihr .wlp Projekt aufteilt. Und zwar werden nicht Rechengebiete (Empfänger) aufgeteilt, sondern die Quellen werden aufgeteilt in Portionen von Quellen. Die Aufteilung der Quellen ermöglicht es, die Resultate sehr einfach zusammenzuhängen. Bei Aufteilung von Lärmkarten wäre das Zusammenfügen komplizierter.

Aufgrund dieser Gegebenheit bringt es beispielsweise nichts, ein Projekt mit nur einer Quelle und einem riesigen Empfangsgebiet parallelisiert laufen zu lassen. So ein Projekt läuft ohnehin nicht besonders lang.

### 5. Start von MultiSonArms unter dem SonArms GUI

Unter dem GUI starten Sie die Berechnung fast wie bisher. Wollen Sie eine Lärmkarte berechnen lassen und wünschen Sie Beschleunigung, so setzen Sie den Haken im Dialog „Immissionspegel berechnen“ bei der neuen Checkbox „Multiprozess“. Ohne Haken läuft alles wie bisher. (Wollen Sie Empfangspunkte berechnen, steht Ihnen keine derartige Beschleunigung zur Verfügung, da normalerweise solche Berechnungen schnell fertig sind.)

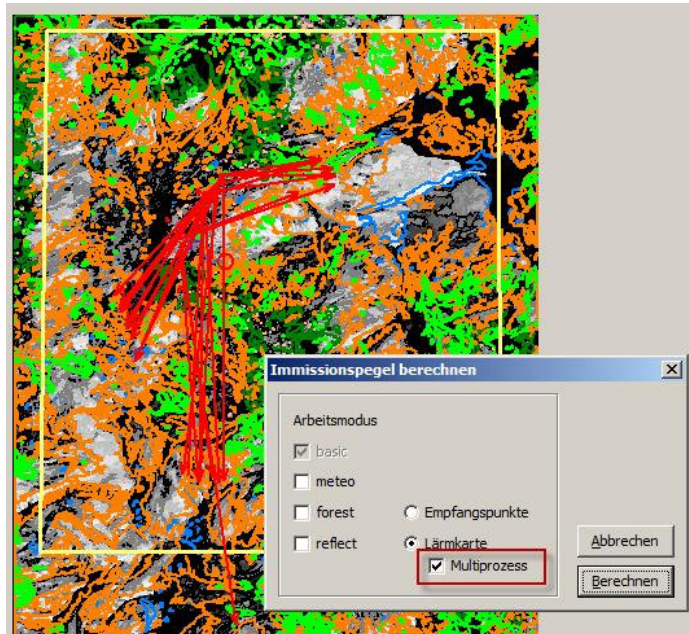


Bild 1: Start der Immissionspegelberechnung

Wenn Sie anschliessend die Schaltfläche „Berechnen“ betätigen, werden mehrere Rechenkerne in eigenen Fenstern gestartet. Dazu ein kleiner Einschub: Die Zahl der Prozesse/Fenster richtet sich nach der Anzahl CPUs der verwendeten Maschine sowie nach dem Parameter **MaxThreads** (Menü „Parameterdatei...“):

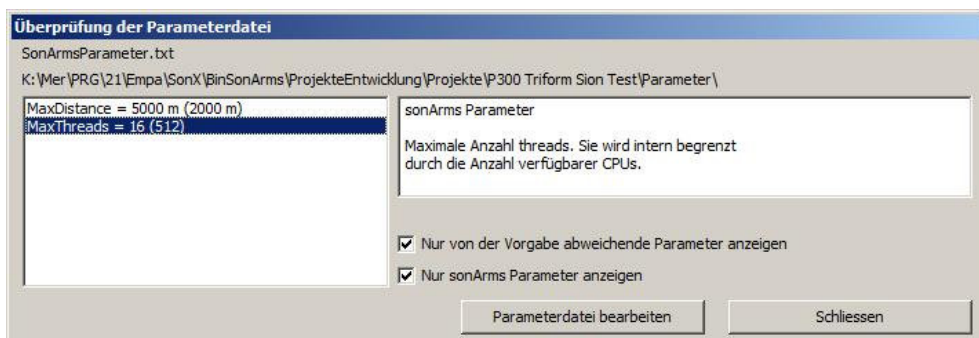


Bild 2: Parameterdatei, Parameter **MaxThreads**

Mit **MaxThreads** legen Sie fest, wieviele Prozesse/Fenster maximal gestartet werden. Ist die Zahl grösser als es CPUs auf Ihrem Rechner hat, begrenzt sie das ausführende Programm auf die Zahl effektiv vorhandener CPUs. Denn mehr Prozesse/Fenster zu haben als es CPUs hat, bringt keine Beschleunigung. Beispiel: Ist **MaxThreads** = 16 und hat es 8 verfügbare CPUs, werden 8 Rechenkern-Fenster gestartet. Der Vorgabewert ist 512, damit ohne Ihren Eingriff unabhängig vom aktuellen Prozessor in jedem Fall alle CPUs genutzt werden.

Weiter mit dem Start: Im Moment wo die Fenster gestartet werden, zieht sich das GUI von der Bildfläche zurück in die Unsichtbarkeit. Zurück bleiben das MultSonArms Anzeigefenster (Bild 3) und dazu die gestarteten Rechenkern-Fenster.

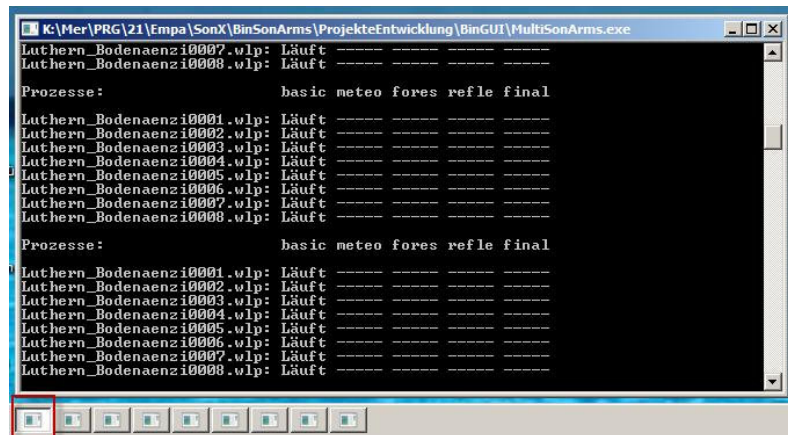


Bild 3: Stand der Prozesse und Taskleiste

Unten in Bild 3 sehen Sie die Taskleiste, die bei Ihnen so oder ähnlich aussieht. Ganz links sehen Sie rot umrandet MultiSonArms, das seine Prozesse alle 10 Sekunden neu anzeigt wie obendran im Bild zu sehen ist. Da sehen Sie für unser Beispiel die untersten 8 Zeilen für die aktuell gestarteten 8 Prozesse, die gerade im basic Modus laufen. Die Zustände der Etappen basic, meteo, forest, reflect werden angezeigt. Die alten Zustandszeilen werden einfach nach oben geschoben.

Wenn Sie die Prozessanzeige in den Vordergrund holen möchten, klicken Sie im Task Manager auf das erste Symbol der angezeigten Reihe. Für die Abschätzung des Zeitbedarfes klicken Sie am besten auf eines der Symbole weiter rechts. Sie zeigen die Rechenkern-Zustände wie Sie sie bisher gewohnt sind.

Ging alles gut, erscheint nach absolvierter Berechnung wie üblich die Botschaft:



Bild 4: Beendet

Darauf stehen Ihnen die Resultate in den wie bisher definierten Verzeichnissen zur Verfügung.

Misslingt auch nur ein Prozess durch Abbruch, so ist die ganze Berechnung misslungen. Auch wenn ein Prozess abbricht, laufen die andern Prozesse möglichst weiter. Dies kann Ihnen helfen, die Ursache des Abbruchs zu finden. Sie können aber auch alles stoppen, wie weiter unten gezeigt wird.

## 6. Resultate und Verzeichnisse

Die Verzeichnisse sind wie bisher üblich, bei unserem Beispiel [P301 Luthern](#) (mit dem Projekt [Luthern\\_Bodenaenzi.wlp](#)), [Input](#), [Log](#), [Out](#), [Parameter](#). Neu kommt ein Arbeitsverzeichnis [TempTasks](#) dazu. Es wird von MultiSonArms gelöscht, erstellt und bearbeitet. Darinnen laufen die Subprojekte/-prozesse. Im Grunde reicht Ihnen dieses Wissen, aber es kann nützlich sein, mehr zu wissen über [TempTasks](#). Das sieht bei unserem Beispiel wie folgt aus:

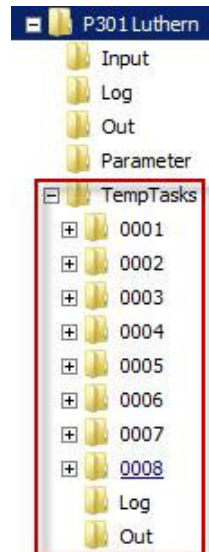


Bild 5: Verzeichnisse

Einzelheiten zu den Verzeichnissen:

- 1) Das Projektverzeichnis ist z.B. [P301 Luthern](#), mit dem Beispielprojekt [Luthern\\_Bodenaenzi.wlp](#).
- 2) Zusammengesetzte Endresultate stehen in [P301 Luthern\Out](#)
- 3) Statusdateien stehen in [P301 Luthern\Log](#)
- 4) MultiSonArms erstellt das Verzeichnis [TempTasks](#). Es befindet sich immer beim .wlp Projekt. Der Name dieses Verzeichnisses ist fix. Es enthält die Umgebungen sämtlicher Subtasks, hier 8 an der Zahl (0001 bis 0008). [TempTasks](#) (mit allen Informationen zu einem vorangegangenen Lauf) wird *beim Start einer Immissionsberechnung mit parallelen Prozessen samt Unterverzeichnissen ohne Rückfrage gelöscht* und neu erstellt. *Speichern Sie also nichts im rot umrandeten Teil, denn sie werden es verlieren!* Sie können dieses Verzeichnis nach den Berechnungen problemlos löschen, wenn Sie die Informationen darin nicht mehr brauchen.
- 5) Die Logdateien aller Subprojekte stehen im Verzeichnis [TempTasks\Log](#). Nur hier erfahren Sie mehr über den Verlauf der Aktivitäten, z.B. über Warnungen. Denn Logdateien werden nicht nach [P301 Luthern\Log](#) kopiert. Sind Ihnen diese Logdateien wichtig für später, müssen Sie sie aus diesem Verzeichnis heraus kopieren, bevor sie beim nächsten Lauf gelöscht werden.
- 6) Die Resultatdateien aller Subprojekte stehen im Verzeichnis [TempTasks\Out](#). Diese setzt das Programm MultiSonArms zusammen und speichert das in [P301 Luthern\Out](#). Normalerweise stehen somit die Resultate an einem sicheren Ort. Bei einem Absturz eines Teilprozesses werden die verbleibenden (intakten) Resultate nicht zusammengesetzt. Dann stehen sie also nur in [TempTasks\Out](#) zur Verfügung und sind beim nächsten Lauf verloren.

## 7. Abbrechen der Berechnungen

Laufen die Berechnungen regulär ab, so beenden sie sich am Ende selbst. Wenn Sie die Berechnungen vor dem regulären Ende abbrechen wollen, so können Sie alle Fenster einzeln schliessen. Es darf keines der angegebenen Fenster offen bleiben, auch nicht das MultiSonArms Fenster. Das würde eine anschliessend gestartete neue Berechnung durcheinander bringen.

Einfacher haben sie es, wenn Sie für einen Abbruch das Batch-Programm [MultiKill.bat](#) starten. Es stoppt alle SonArms Prozesse und auch MultiSonArms sofort, *ohne Rückfrage*. Sie finden das Programm im [TempTasks](#) Verzeichnis beim zugehörigen Projekt, nicht im GUI.

Detail: Starten Sie nur das zum Projekt gehörige [MultiKill.bat](#), nicht eines, das bei einem anderen Projekt gespeichert ist. Denn wenn dort ein anderer Rechenkern benutzt worden ist, funktioniert [MultiKill.bat](#) nicht. Denn [MultiKill.bat](#) wurde beim Start der Berechnungen von MultiSonArms passend erzeugt.

Im selben [TempTasks](#) Verzeichnis befindet sich noch ein anderes Batch-Programm [MultiRun.bat](#). Es startet den Rechenkern mehrfach und parallel. Es wird von MultiSonArms erzeugt und auch aufgerufen.

Normalerweise benutzen Sie das nicht. Es lässt zwar das Projekt parallelisiert laufen, setzt aber die Resultate nicht zusammen. Haben Sie es versehentlich gestartet, hilft [MultiKill.bat](#).

## 8. Resultatunterschiede sind kein Fehler

Eigentlich sollte die parallele Ausführung von Berechnungen gegenüber der herkömmlichen seriellen Ausführung keinen Einfluss auf die Resultate haben, würde man denken. Eingangs wurde jedoch darauf hingewiesen, dass die serielle Ausführung der Berechnungen und die parallelisierte Variante zu geringfügig unterschiedlichen Ergebnissen kommen können. Wir sehen zwei Gründe, warum dies eintreten kann:

- 1) **Ausscheidung von Gebäudeflächen.** Der reflect Modus scheidet Gebäudeflächen aus, die weit weg von allen Empfängern des Projektes und weit weg von allen Quellen (Mündung und Ziel) des Projektes sind. Infolge der Aufteilung der Quellen in Portionen decken die beteiligten Quellen im Subprojekt ein kleineres Gebiet ab als im Originalprojekt. Dadurch wird auch das Gebiet der berücksichtigten Reflektoren kleiner werden. Das lässt sich etwas abfangen durch die Vergrößerung des Parameters [ReflMaxDist](#). Da es sich um weit entfernte Reflektoren handelt, ist dieser Einfluss aber eher gering.
- 2) **Umgang des Gleitkommaprozessors mit den hintersten Dezimalstellen.** Diese ausgetüftelte Technik hilft, dass gebrochene Zahlen wieder ganz werden können. Läuft ein Programm im *single thread Modus* oder im *multi thread Modus*, so ist der Umgang des Prozessors mit diesen kleinsten Zahlen *etwas verschieden*, wie unsere Untersuchungen gezeigt haben.  
Einzelheiten:

Je nachdem wie der Haken in der Checkbox „Multiprozess“ (Bild 1) gesetzt ist gilt:

Haken gesetzt: Die parallelen Subprozesse laufen für sich im *single thread Modus*, mit lokaler Einstellung [MaxThreads](#) = 0.

Haken nicht gesetzt: Intern im SonArms Rechenkern laufen einige Routinen im *multi thread Modus* ab (was basic und meteo etwas beschleunigt). Mit [MaxThreads](#) > 0.

Während dies für den *basic*, *forest* und *reflect* Modus zu keinen nennenswerten Resultatunterschieden führt, kann es im *meteo* Modus in Einzelfällen zu Unterschieden von mehreren dB in einem Einzelresultat kommen. Der Grund liegt darin, dass der meteo Modus mit seinem Raytracing sehr empfindlich für kleinste Unterschiede („Schmetterlings-Flügelschlag“) sein kann, sodass ein Entscheid kippen kann. Solch zufällige Resultatunterschiede sollten niemanden stören, denn man kann das auch als Bild nehmen für die Labilität der Ausbreitung bei verschiedenen Wetterbedingungen. Da normalerweise aufgrund der Wetterstatistiken mehrere Meteosituationen beteiligt sind, sollten sich solche Unterschiede mehr oder weniger ausgleichen.

## 9. Referenzdokumente

- [1] SonArms Dokumentation im SonArms.zip Paket à Google mit „sonArms“ oder direkt:  
<https://www.bafu.admin.ch/bafu/de/home/themen/laerm/fachinformationen/laermmittlung-und-beurteilung/ermittlung-und-beurteilung-von-schiesslaerm.html>
- [2] Anleitung zum Gebrauch von MultiSonArms 23-Feb-22.doc