

Anleitung zum Excel-Addin ESCMA.xla

Abdelkarim Habib
Martin Wilke
Marcel Neitsch
Iván Santibáñez Koref*

19. März 2006

*Fehlermeldungen und Kommentare an: iskbionik.tu-berlin.de

Version 1.0 , March 2006

Copyright © 2006 Habib, Neitsch, Santibáñez Koref, Wilke (HNSW)

Fachgebiet Bionik und Evolutionstechnik
Technische Universität Berlin
Skr. ACK1
Ackerstrasse 71-76
13355 Berlin

Fax.: (+49 30) 314 72 019
Tel.: (+49 30) 314 72 655
Email: isk@bionik.TU-Berlin.DE

Inhaltsverzeichnis

1	The GNU General Public License	4
2	Einleitung	9
3	Theorie der Evolutionsstrategie	10
3.1	Die (μ, λ) -Evolutionsstrategie	12
3.2	Die $(\mu/\mu, \lambda)$ -Evolutionsstrategie	12
3.3	Die Kovarianz - Matrix - Adaption(CMA) [2]	12
4	Ausführliche Anleitung zur Benutzung und Funktion des Excel-Addins	14
4.1	Basiseinstellungen	14
4.2	Erweiterte Einstellungen	18
4.3	Beschreibung der Schnittstelle zur Implementierung in andere Makros . .	20
5	Kurzanleitung	21

Abbildungsverzeichnis

1	Ablaufdiagramm einer einfachen (1,3) - ES	11
2	Kreisförmige und Ellipsoide Mutationsverteilung	13
3	Untermenü zur Einstellung der Makrosicherheit in Excel	14
4	Eingetragene Messwerte im Excelsheet mit geplotteter Funktion.	15
5	Die Modellfunktion wird für den ersten Messwert definiert.	16
6	Die fertige Beispieltabelle, mit der die Koeffizienten angepasst werden können.	17
7	Basiseinstellungen des ES-Optimizers	18
8	Die fertige Beispieltabelle, nach der Anpassung der Koeffizienten.	19
9	Erweiterte Einstellungen „Stopping Criteria”	20
10	Erweiterte Einstellungen „Generation Properties”	21
11	Screenshot einer Beispieldatei in Excel zur Verwendung des Addins	22

1 The GNU General Public License

Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author’s protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program”, below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
 - a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
 - b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
 - c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
 - a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

one line to give the program’s name and a brief idea of what it does.
Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) yyyy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’.
This is free software, and you are welcome to redistribute it under certain conditions; type ‘show c’ for details.

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
‘Gnomovision’ (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

2 Einleitung

Als Ingenieur steht man häufig vor dem Problem an einen Satz gemessener Datenpunkte eine Modellfunktion anpassen zu müssen. Solange diese linear ist, führen mathematisch analytische Methoden schnell ans Ziel. Das Prinzip beruht darauf ein Minimalwertproblem zu lösen. Hierzu wird eine Fehlerfunktion (Summe der Fehlerquadrate) definiert, in die die Modellfunktion eingesetzt wird. Bei linearen Funktionen führt diese Vorgehensweise letztendlich dazu ein lineares Gleichungssystem zu lösen. Sind die Daten jedoch nur mit einer nichtlinearen Funktion anzupassen, kann das genannte Minimierungsproblem nur iterativ angenähert werden. Die klassische Mathematik schlägt dafür lokal deterministische Methoden, wie z.B. die Gradientenstrategie vor. Das hier beschriebene Excel-Addin hingegen verwendet zur Lösung dieser Aufgabe die Evolutionsstrategie. Der Vorteil liegt darin, dass der Fortschritt der Evolutionsstrategie bei hoher Anzahl der Problemdimensionen schneller vonstatten geht als bei der Gradientenstrategie. Je komplizierter also die Modellfunktion ist, desto empfehlenswerter ist es die Evolutionsstrategie zu verwenden.

Das vorliegende Excel-Addin ermöglicht es auf einfache Art und Weise eine Modellfunktion für einen bestimmten Datensatz mit oben beschriebener Methode anzupassen. Mit Hilfe der grafischen Funktionen von Excel kann auf diese Weise schnell überprüft werden, ob der gewählte Ansatz die Daten hinreichend genau annähert. **This addin may be distributed under the terms of General Public License (Seite 5: Terms and Conditions For Copying, Distribution and Modification).**

Die Vorliegende Anleitung gliedert sich in drei Teile. Im ersten Abschnitt wird auf die Theorie der Evolutionsstrategie(ES) eingegangen. Es wird die allgemeine Vorgehensweise einer einfachen Strategie dargelegt, sowie die grundlegenden Strategieparameter erläutert. Weiterhin wird kurz auf die Funktionsweise, der im Programm verwendeten höheren ES eingegangen. Dieser Abschnitt enthält keine wichtigen Informationen um das Addin in seiner Grundfunktionalität zu bedienen. Wer also nur schnell ein vorliegendes Problem lösen will, kann diese Seiten ruhig überblättern. Sie dienen lediglich dazu dem Leser einen groben Überblick zu verschaffen nach welchen Algorithmen das Programm prinzipiell arbeitet.

Der zweite Abschnitt befasst sich mit der genauen Bedienung des Addins. Anhand eines Beispiels wird ausführlich erläutert wie das Addin zu verwenden ist um einen gegebenen Datensatz zu „fitten“. Weiterhin werden alle möglichen Einstelloptionen erklärt. Dieses Kapitel ist an diejenigen gerichtet, die praktisch schnell ein Problem lösen wollen aber nicht viel Erfahrung im Umgang mit Excel Makrofunktionen und/oder der Evolutionsstrategie haben.

Für alle, die es besonders eilig haben, gibt der letzte Teil nochmal eine Anleitung zur Bedienung in absoluter Kurzform. Wer sich mit Excel und der Evolutionsstrategie auskennt, wird hier am schnellsten zum Ziel gelangen.

3 Theorie der Evolutionsstrategie

Die Evolutionsstrategie ist eine numerische Optimierungsmethode die nach dem Vorbild der biologischen Evolution aufgebaut ist. Diese arbeitet allgemein nach dem Darwinschen Prinzip, nach dem nur die Besten überleben: Das bedeutet genauer gesagt, dass nur diejenigen Individuen einer Population überleben, die sich in der vorliegenden natürlichen Umgebung am besten behaupten können. Nur diese Individuen übertragen ihr Erbmaterial auf die Folgegeneration, womit über den Lauf der Jahre gute Erbeigenschaften akkumuliert werden, während schlechte sozusagen „aussterben“. Dieser sehr einfache Grundalgorithmus der Evolution lässt sich recht einfach auf die Technik übertragen. Wichtig ist dabei zunächst eine Qualitätsfunktion zu definieren, die es erlaubt die Individuen untereinander zu vergleichen, um letztendlich eine Aussage darüber machen zu können welches sich in der vorgegebenen Umgebung am besten behauptet. Diese Qualitätsfunktion hängt natürlich davon ab, welches Ziel man mit der Optimierung erreichen will. Die einzelnen Individuen besitzen, genau wie in der Natur, gewisse Erbmerkmale, in diesem Fall durch Variablen dargestellt, die unterschiedlich ausgeprägt sein können. Diese Ausprägung ist durch den konkreten Wert der jeweiligen Variable gegeben. Ein Individuum wird mathematisch also durch einen Vektor dargestellt. Je mehr Eigenschaften es besitzt, desto höher ist die Dimension des Vektors.

Die einfachste denkbare Strategie ist eine so genannte $(1,\lambda)$ -ES. Bei dieser Vorgehensweise wird zunächst ein Elter erzeugt. Das bedeutet, dass jedes Merkmal für den Anfang zufällig bestimmt wird. Dieser Elter setzt λ Kinder in die Welt. Mathematisch bedeutet dies, dass jedes Kind wie folgt aus dem Eltervektor berechnet wird:

$$\overrightarrow{X}_{Kind} = \overrightarrow{X}_{Elter} + \overrightarrow{Z}_{gauss} \cdot \delta \quad (1)$$

$\overrightarrow{Z}_{gauss}$ ist ein Vektor, dessen Einträge $(0,1)$ -Normalverteilte Zufallszahlen sind. Diese Wahrscheinlichkeitsverteilung beschreibt die natürliche Mutation am Besten: geringe Abweichungen treten grundsätzlich häufiger auf als starke. δ ist ein Skalar und stellt die Schrittweite dar. Sie konditioniert im Prinzip die Mutation auf das vorliegende Problem, gibt also an wie stark die Merkmale der Kinder im Mittel von denen des Elters abweichen. Sind alle λ Nachkommen erzeugt worden, wird mit Hilfe der Qualitätsfunktion der beste Nachkomme bestimmt. Diese Bestimmung kann im Computer durch direkte Auswertung einer Funktion erfolgen. Sie kann aber auch dadurch gemacht werden, dass jedes Individuum technisch in irgendeiner Art realisiert wird, einem Experiment unterzogen und der Ausgang dieses Experiments wiederum in eine Qualitätsfunktion eingerechnet wird. Die Definition der Qualität ist letztendlich ausschlaggebend. Ist der beste Nachkomme der ersten Generation bestimmt worden, wird er als Elter der nächsten Generation gesetzt. Die Schritte wiederholen sich dann solange bis kein Fortschritt mehr stattfindet und der Lauf abgebrochen wird.

Es sei hier erwähnt, dass dies nur die absoluten Grundzüge der Theorie sind. Für das sichere Funktionieren der Strategie ist noch mindestens eine Steuerung der Schrittwei-

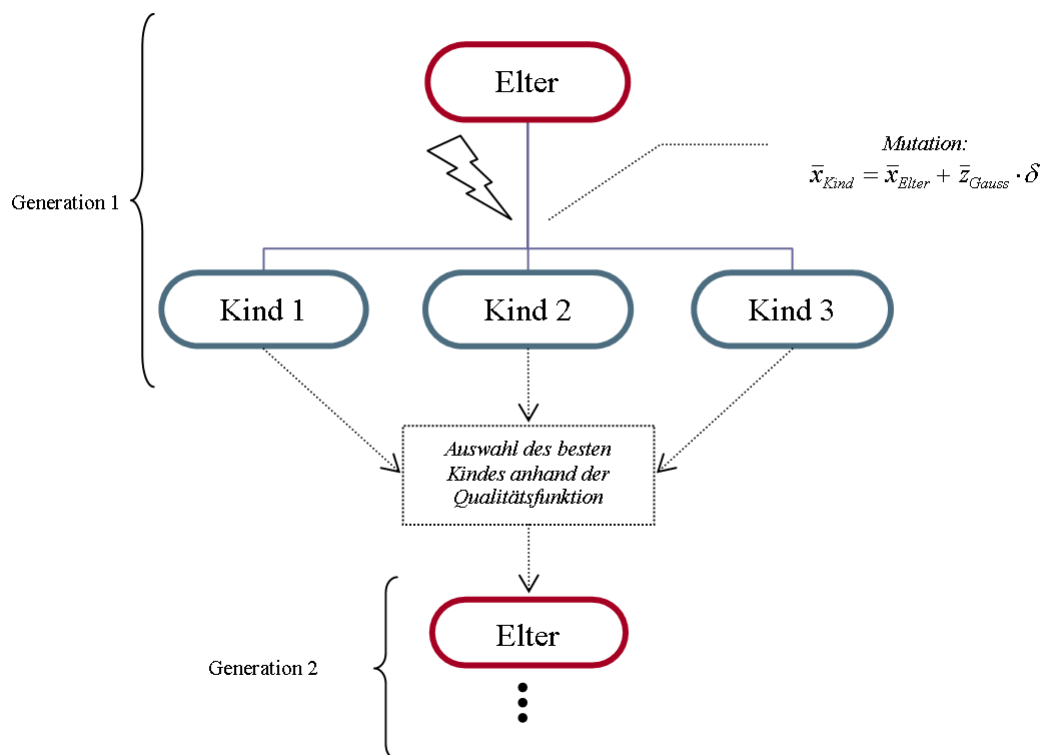


Abbildung 1: Ablaufdiagramm einer einfachen (1,3) - ES

te (z.B. durch „mutative Schrittweitenregelung“) notwendig, auf die hier nicht weiter eingegangen wird. Interessierten Lesern wird weiterführende Literatur [1] empfohlen.

3.1 Die (μ, λ) -Evolutionstrategie

Die (μ, λ) -ES unterscheidet sich nicht sehr stark von der $(1, \lambda)$ -ES. Wie zu erwarten ist, arbeitet erstere Strategie nicht mit einem, sondern mit μ Eltern. Um einen Nachkommen zu bestimmen, wird zufällig (gleichwahrscheinlich) einer der μ Eltern ausgewählt und wie im vorigen Abschnitt beschrieben mutiert. Nachdem alle λ Kinder erzeugt wurden, werden die besten μ Kinder zu den Eltern der nächsten Generation.

Vorteile hat die (μ, λ) -ES gegenüber der $(1, \lambda)$ -ES hauptsächlich in sogenannten „verrauschten Qualitätsgebirgen“, in denen die Bewertung der Qualität eines Individuums einer gewissen Streuung unterworfen ist (siehe [1] S.190-191). Solche Voraussetzungen sind jedoch bei reinen Computerexperimenten, wie sie das vorliegende Programm durchführt, gewöhnlicher Weise nicht gegeben.

3.2 Die $(\mu/\mu, \lambda)$ -Evolutionstrategie

Eine $(\mu/\mu, \lambda)$ -ES bezieht die Wirkung der sexuellen Fortpflanzung auf die Evolution in ihren Algorithmus mit ein. Wie bei der vorher erklärten Strategie erzeugen μ Eltern λ Kinder. Die Mechanismen, die hierbei angewendet werden unterscheiden sich jedoch grundsätzlich voneinander. Anstatt einen Elter zufällig zur Mutation auszuwählen, werden die einzelnen Erbmerkmale aller Eltern arithmetisch gemittelt. Es entsteht damit ein sogenannter Elternschwerpunkt, der nun wiederum wie bereits bekannt λ mal mutiert wird um die einzelnen unterschiedliche Kinder zu erhalten.

Die Bezeichnung des beschriebenen Vorgangs wird in der Evolutionstrategie „Kontinuums - Rekombination“ genannt (siehe [1] S.139-140). Andere Merkmalsvermischungen wie die diskrete Rekombination, werden hier nicht weiter betrachtet. Die Vorteile von rekombinativen Strategien kommen ebenfalls besonders deutlich in verrauschten Qualitätsgebirgen zu tragen. Sie haben noch zusätzlich den Vorteil bei großer Variablenzahl wesentlich höhere Fortschrittsgeschwindigkeiten als eine $(\mu/\mu, \lambda)$ -ES zu besitzen.

3.3 Die Kovarianz - Matrix - Adaption(CMA) [2]

Der Algorithmus, der in dem vorliegenden Programm implementiert ist, wird als Kovarianz - Matrix - Adaption bezeichnet verhält sich aber im Prinzip wie eine $(\mu/\mu, \lambda)$ -ES. Das Wirken einer CMA-ES bezieht sich nämlich prinzipiell nur auf die Mutationsverteilung der Nachkommen. Die Mutation eines Elternvektors wird in einer gewöhnlichen ES mathematisch laut Formel 1 beschrieben. Eine derartige Erzeugung der Nachkommenvektoren im hochdimensionalen Raum führt dazu, dass diese im Mittel auf der Schale einer Hyperkugel, deren Mittelpunkt der Elter bildet, angeordnet werden. Die Schrittweite δ beschreibt dabei den Radius dieser Hyperkugel.

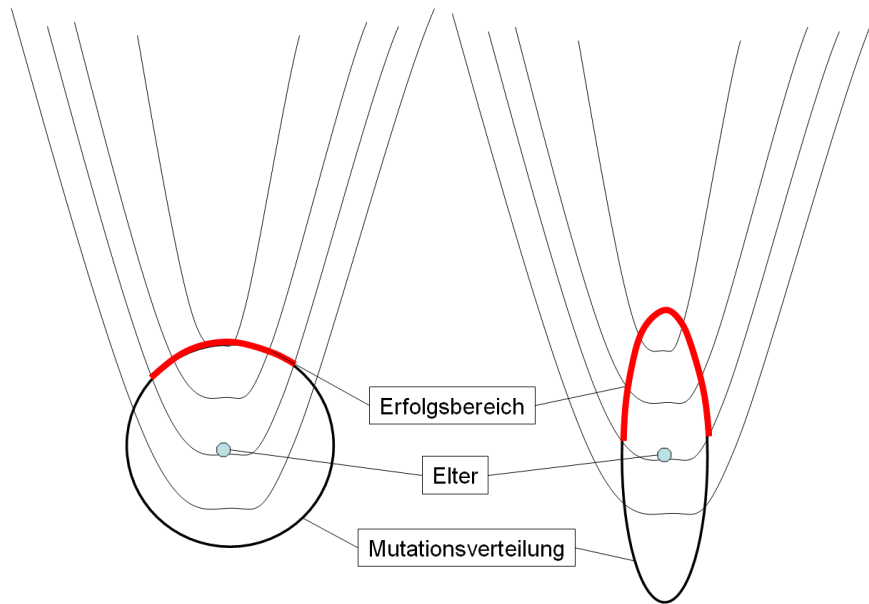


Abbildung 2: Kreisförmige und Ellipsoide Mutationsverteilung

Betrachtet man Abbildung 2, wird klar dass es durchaus sinnvoll sein kann die Mutationsverteilung entsprechend der Qualitätslandschaft anzupassen. Die Informationen, die hierfür benötigt werden zieht die CMA-ES aus den selektierten Nachkommen der letzten Generationen. Gestartet wird beispielsweise von einer kreisrunden Mutationsverteilung. Ist der Elternschwerpunkt der laufenden Generation bestimmt, wird die Verteilung in seine Richtung etwas gestreckt. Da weiter zurückliegende Generationen vernünftiger Weise einen geringeren Beitrag zur aktuellen Mutationsverteilung haben sollten, werden die Schwerpunkte je länger sie von der aktuellen Generation zurückliegen abgeschwächt gewichtet. Auf diese Weise passt sich die Verteilung der Nachkommen immer der lokalen Form des Qualitätsgebirges an.

Die beschriebene Vorgehensweise erklärt zwar den Mechanismus der Anpassung sehr gut, ist aber nicht ratsam genau so zu implementieren. Jeder während der Evolution erzeugte Elternschwerpunkt müsste gespeichert werden, wodurch der Speicherplatzbedarf exponentiell mit der Anzahl der Generationen g ansteigen würde. Stattdessen wird die Mutationsverteilung durch Anpassung ihrer Kovarianzmatrix mit Hilfe sogenannter Evolutionspfade angepasst. Der Speicherplatzbedarf des Algorithmus beschränkt sich somit auf $O(n^2)$.

4 Ausführliche Anleitung zur Benutzung und Funktion des Excel-Addins

Dieser Abschnitt teilt sich in drei Unterabschnitte auf. Als erstes sollen die Basiseinstellungen beschrieben werden die für die Durchführung einer jeden Berechnung zwingend notwendig sind. Anschliessend werden die erweiterten Einstellungen vorgestellt die es einem ermöglichen die Parameter der verwendeten ES selber einzustellen. Zu guter Letzt erfolgt eine Vorstellung der Schnittstelle des Addins um eine Implementierung in eigenen Quellcode zu vereinfachen.

4.1 Basiseinstellungen

Die Einstellung die auf jeden Fall erfolgen muss um das Addin benutzen zu können ist die Anpassung der Sicherheitseinstellungen von Excel. Hierzu muss das Untermenü Extras → Makro → Sicherheit aufgerufen werden (Abb. 3). Es erscheint dann ein Fenster in

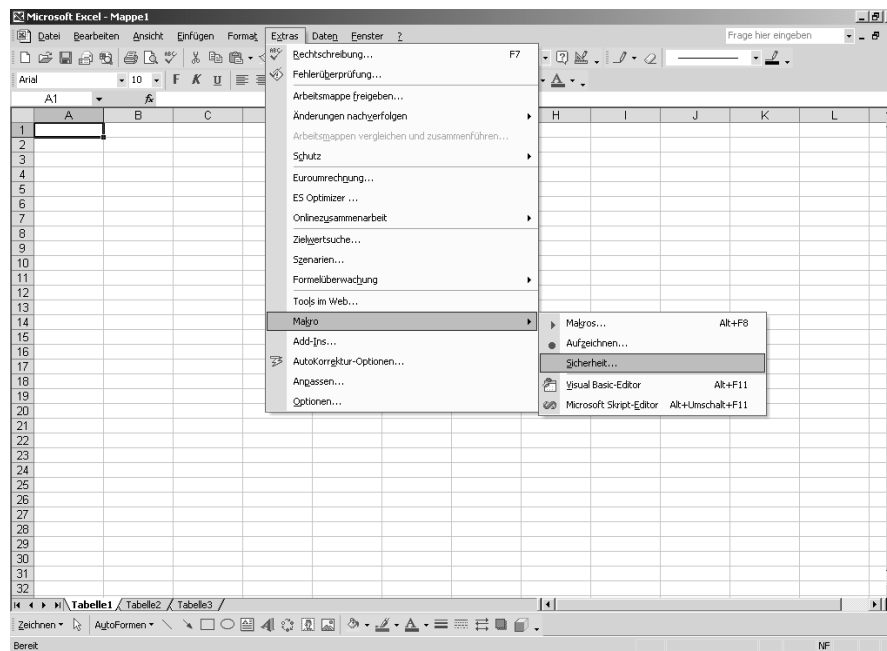


Abbildung 3: Untermenü zur Einstellung der Makrosicherheit in Excel

dem die Einstellungsstufe auf Mittel gestellt werden muss. Ist die Stufe „Mittel“ gewählt, muss man nach dem Öffnen des Makros einmal bestätigen, dass man das Makro wirklich ausführen will. Bei der Einstellung „Niedrig“ entfällt diese Abfrage. Excel führt dann allerdings auch ungefragt JEDES Makro aus. Es ist ausserdem möglich den ES-Optimizer Standardmässig beim Start von Excel mitzuladen. Dazu muss man im Untermenü Extras → Add-Ins den esma.xla Makro eintragen.

Sind diese Voreinstellungen erfolgt, kann man den Sheet mit den Messdaten öffnen. In dieser Anleitung soll die Erklärung des Makros anhand eines Beispiels durchgeführt werden. Man öffnet also seine Messdaten in Excel oder gibt sie in einen leeren Sheet ein. Anschließend erstellt man ein Diagramm aus den Daten um einen optischen Eindruck des Funktionsverlaufes zu bekommen. Abbildung 4 zeigt eine Beispieltabelle, in der diese Arbeitsschritte bereits durchgeführt wurden. Der Funktionsgraph erinnert an die Gauß-

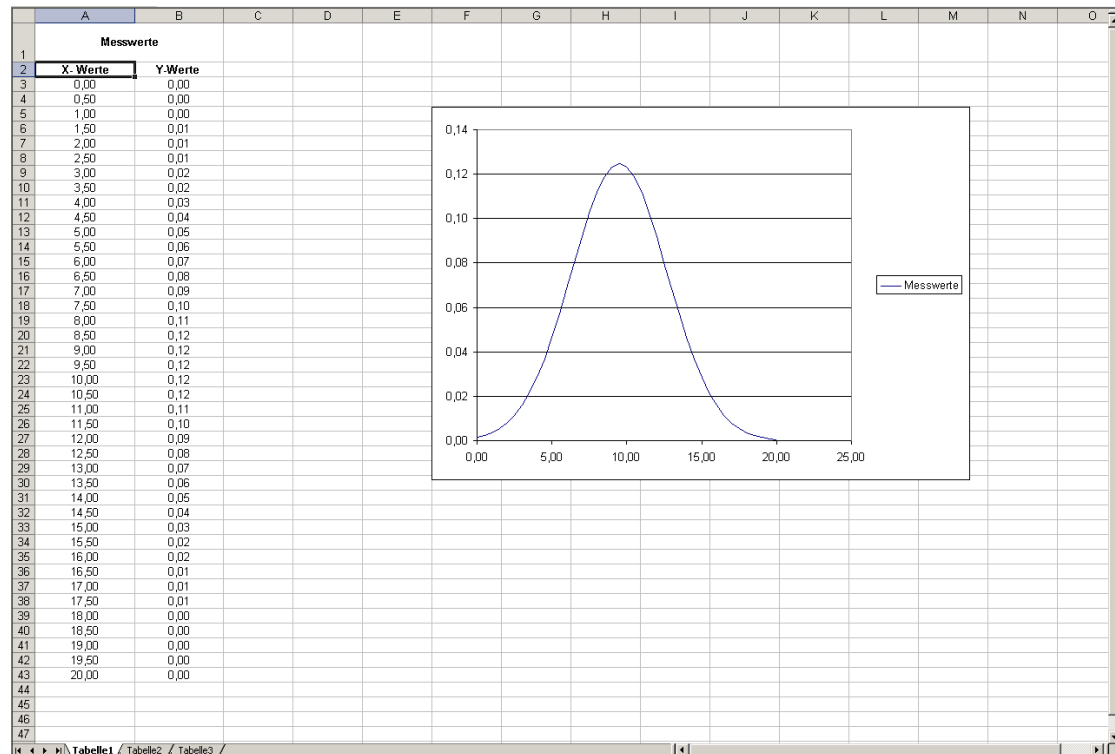


Abbildung 4: Eingetragene Messwerte im Excelsheet mit geplotteter Funktion.

sche Fehlerfunktion, weshalb diese im weiteren Verlauf als Modellfunktion verwendet werden soll. Die Gaußsche Fehlerfunktion ist definiert als:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Sie hat die zwei Koeffizienten σ und μ , die jetzt dahingehend angepasst werden sollen, dass die Funktion bei den gegebenen X-Werten möglichst die gleichen Y-Werte produziert, wie sie durch die Messwerte gegeben sind. In der Wahrscheinlichkeitstheorie wird μ der Erwartungswert genannt, während σ die Streuung angibt. Theoretisch könnte jede beliebige Modellfunktion an die Messwerte angepasst werden. Welche Funktion sinnvoller Weise gewählt werden sollte, liegt im Ermessen des Benutzers.

Um σ und μ mit Hilfe des Makros anzupassen, muss jetzt die Funktion auf die durch die Messdaten vorliegenden X-Werte angewendet werden. Dazu definiert man zunächst

zwei Zellen, die für die beiden Koeffizienten stehen und trägt zwei beliebige Startwerte ein. Anschließend trägt man in die Zelle neben dem ersten Messwert die Gaussfunktion ein, wobei die Parameter σ und μ als feste Zellbezüge angegeben werden, während der entsprechende X-Wert als relativer Zellbezug angegeben wird. In Zelle C2 steht also letztendlich:

$$=((\text{WURZEL}(2*\text{PI}())*\$F\$3)^{-1})*\text{EXP}(-((A3-\$G\$3)^2)/(2*(\$F\$3)^2))$$

Die bis dahin erstellte Tabelle ist in Abbildung 5 zu sehen. Die Zelle C2 kann nun

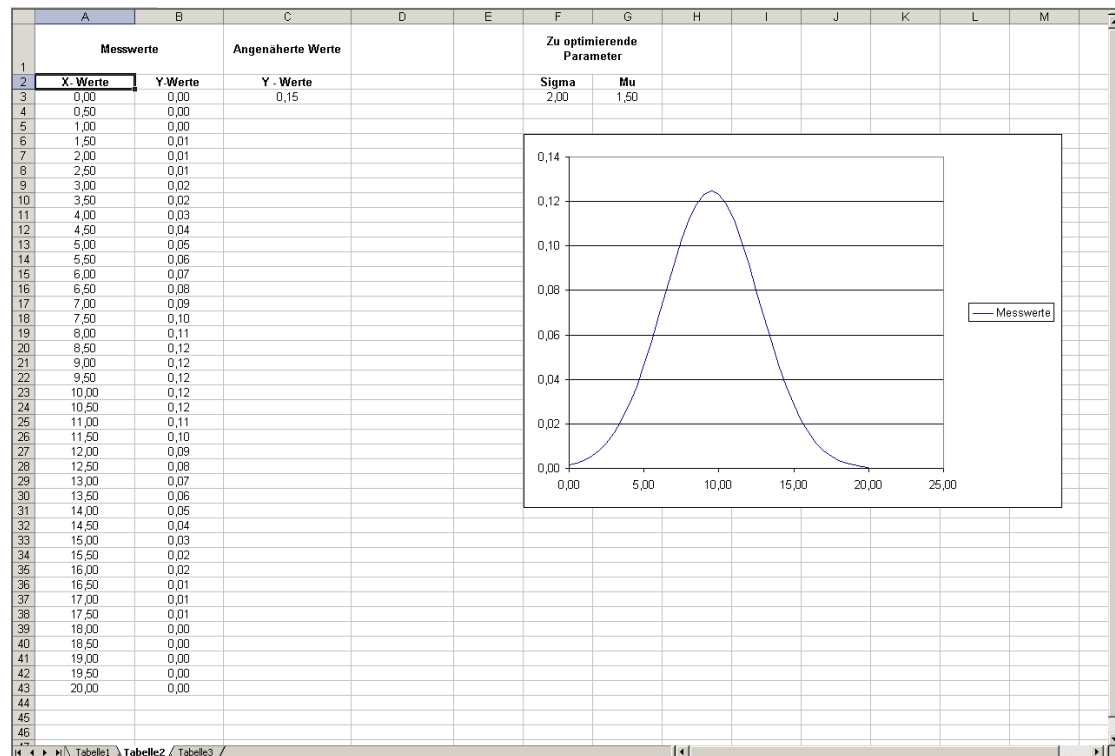


Abbildung 5: Die Modellfunktion wird für den ersten Messwert definiert.

kopiert werden, so dass neben jedem Y1-Messwert ein mit Hilfe der Modellfunktion berechneter Y2-Wert steht. In die Grafik wird eine weitere Datenreihe (X,Y2) hinzugefügt. Es muss nun noch eine Qualitätsfunktion erstellt werden damit die Evolutionsstrategie funktionieren kann. Dazu wird für jedes Messwert/Näherungswert - Paar das Quadrat der Abweichung berechnet. Anders ausgedrückt wird für jeden Messwert der vorzeichenfreie Fehler $err = (Y1 - Y2)^2$ gebildet. Abschließend werden alle diese Fehler zu einem Gesamtfehler aufsummiert. Die fertige Tabelle ist in Abbildung 6 zu sehen.

Man hat nun alle notwendigen Vorbereitungen getroffen damit die Anpassung der Koeffizienten durch die Evolutionsstrategie mit Hilfe des Makros erfolgen kann. Dazu wählt man im Menü den Punkt Extras \mapsto ES-Optimizer worauf sich ein Dialogfenster

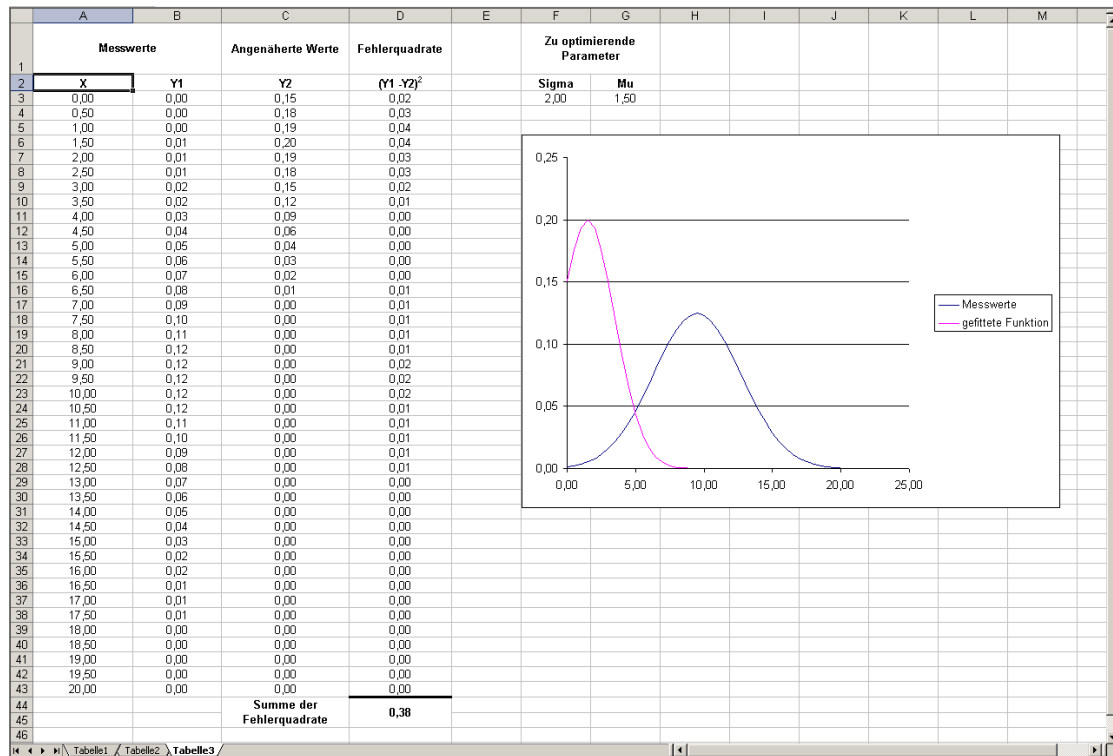


Abbildung 6: Die fertige Beispieldaten, mit der die Koeffizienten angepasst werden können.

öffnet (siehe Abb. 7) in dem man die Basiseinstellungen vornehmen muss. Wie man an der Abbildung schon erkennen kann, ist der ES-Optimizer aktuell nur für Minimierungsprobleme ausgelegt. Die erforderlichen Einträge für die zwei vorhandenen Eingabefelder werden im Folgenden erklärt.

- Parameters to fit: Die Zellen in denen die Werte stehen die angepasst werden sollen. Also die Zellen in denen die Koeffizienten der anzupassenden Funktion stehen. In unserem Beispiel sind das die zwei Zellen F2 (σ) und G2 (μ).
- Function to optimize: Die Zelle in der der zu minimierende Wert steht. In unserem Beispiel also die Zelle, in der die Summe der Fehlerquadrate steht.

Sind diese Einstellungen erfolgt, wird die Berechnung durch Anklicken des Start-Buttons gestartet. Nach erfolgreicher Berechnung wird die Anzahl der Iterationen oder in evolutionsstrategischer Sprache die Anzahl der Generationen angezeigt und die neuen angepassten Koeffizienten eingetragen. Durch die grafische Darstellung lässt sich jetzt schnell erkennen ob die Messwerte von der neuen errechneten Kurve gut nachgebildet werden

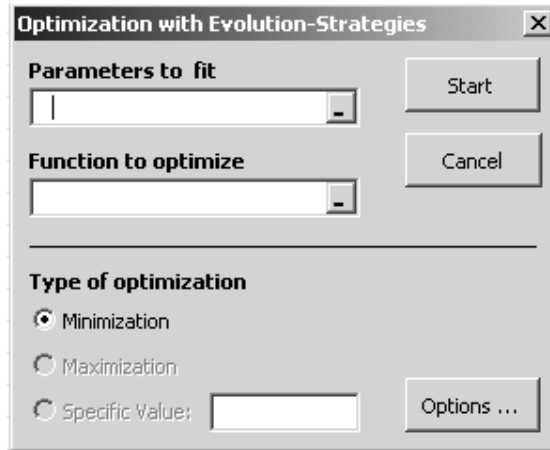


Abbildung 7: Basiseinstellungen des ES-Optimizers

oder nicht. Im vorliegenden Beispiel (Abb.8) sind die Messwerte und die Aproximierungskurve fast Deckungsgleich.

4.2 Erweiterte Einstellungen

Klickt man auf das Bedienfeld „Options“ so erhält man die Möglichkeit erweiterte Einstellungen vorzunehmen die hier im Detail erklärt werden sollen.

Es werden 2 Registerkarten („Stopping Criteria“ und „Generation Properties“) angezeigt die die erweiterten Einstellung beinhalten.

- Stopping Criteria (siehe auch Abb. 9)
 - Stopfitness: Der Wert der zu fittenden Funktion (Qualitätsfunktion), bei dem die Optimierung abgebrochen wird. Gewöhnlicher Weise wird der optimale Wert Null der Qualitätsfunktion (z.B.: $Q = \Sigma (x_{berechnet} - x_{ziel})^2$) nicht exakt erreicht. Die Stopfitness gibt in diesem Fall die Abweichung vom optimalen Wert an, die der Benutzer als ausreichend klein ansieht, und somit die Optimierung abgebrochen werden kann. Die Stopfitness wird auch als Abbruchkriterium herangezogen, wenn der Fortschritt der Optimierung zu kleine Werte annimmt. Dazu wird aus einer bestimmten Anzahl der letzten Generationen, die Differenz der besten und schlechtesten Qualität gebildet und durch die zuletzt berechnete Qualität geteilt. Ist das Ergebnis kleiner als die definierte Stopfitness wird der Optimierungsprozess angehalten und der Benutzer kann sich entscheiden, ob weiter fort gefahren werden soll. Die Anzahl der Generationen, die bei dieser Berechnung mit einbezogen werden, ist in der Variable: „Number of Generations for Calculating Precision“ festgelegt.

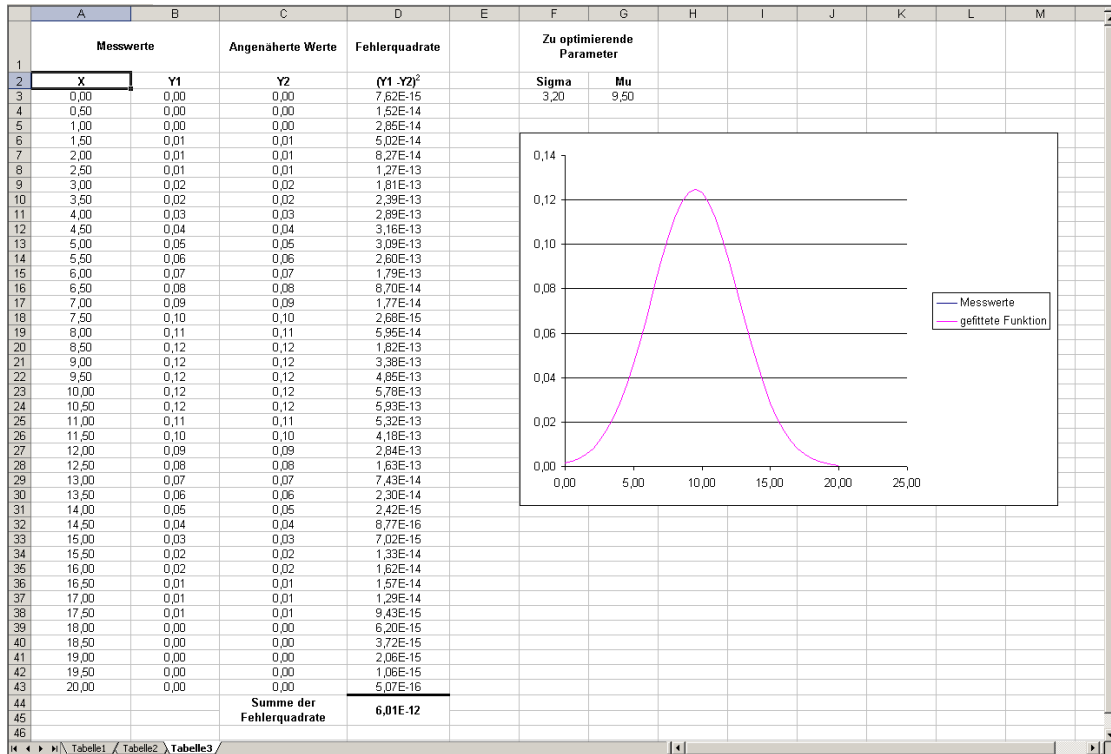


Abbildung 8: Die fertige Beispieltabelle, nach der Anpassung der Koeffizienten.

- Number of Generations for Calculating Precision: Die Anzahl der Generationen, die bei der Berechnung des Vergleichswertes für die Stopfitness herangezogen werden soll. (Siehe auch unter „Stopfitness“)
- Maximum Number of Function Evaluations: Legt die maximale Anzahl der Auswertungen der Qualitätsfunktion fest, nach der die Optimierung spätestens abgebrochen wird.
- Generation Properties: (siehe Abb. 10)
 - Der implementierte Algorithmus arbeitet prinzipiell nach einer $(\mu/\mu, \lambda)$ - Evolutionsstrategie. Das heißt, es werden in jeder Generation λ Kinder erzeugt, von denen die μ besten ausgewählt werden. Die Eigenschaften dieser μ Elter, werden rekombiniert, so dass ein „Elterschwerpunkt“ entsteht, von dem aus die neuen λ Kinder durch Mutation erzeugt werden.
 - Number of Parents: gibt die Anzahl der Elter an.
 - Number of Offspring: Die Anzahl der Kinder, die pro Generation erzeugt werden. (Siehe auch unter „Number of Parents“)
 - Initial Global Step Size: Die Kinder einer Generation werden durch Mutation erzeugt. Dies geschieht dadurch, dass auf den Variablensatz, der das Elter -

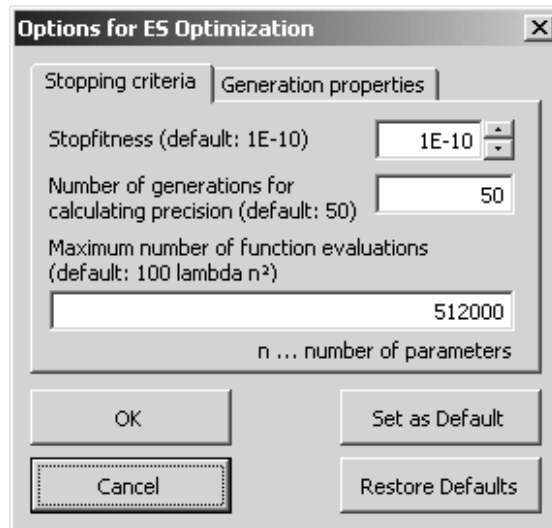


Abbildung 9: Erweiterte Einstellungen „Stopping Criteria“

Individuum repräsentiert, ein normalverteilter Zufallsvektor $N(0,1)$ addiert wird. Dieser Zufallsvektor wird wiederum mit einem konstanten Faktor, der Schrittweite multipliziert. Bei der „Initial Global Step Size“ kann der erste zu verwendende Wert dieser Schrittweite eingestellt werden. Die Schrittweite wird im Laufe der Optimierung angepasst.

4.3 Beschreibung der Schnittstelle zur Implementierung in andere Makros

Das Add-In `escma.xla` besitzt ein Modul, welches dem Benutzer ermöglicht den Algorithmus bequem direkt von anderen Makros aufzurufen. Das Modul hat den Namen „SubroutineWithoutDialog“ und stellt zwei Subroutinen zur Verfügung. Beide sollten zweckmäßiger Weise mit benannten Parametern aufgerufen werden, die selbsterklärend sind wenn man mit der Bedienung des Makros vertraut ist. Es folgen die Namen der Makros, mit den Namen der jeweiligen Parameter. Die Klammern hinter den Parameternamen geben an, ob das Argument optional ist. Wird ein optionales Argument nicht explizit angegeben, verwendet das Add-in den entsprechenden Standardwert:

- OptionsESCMA
 - NoOfOffspring (*optional*)
 - NoOfParents (*optional*)
 - StopFitness (*optional*)
 - NoOfGenerationsForStopFitness (*optional*)
 - MaxNoOfFunctionEvaluations (*optional*)

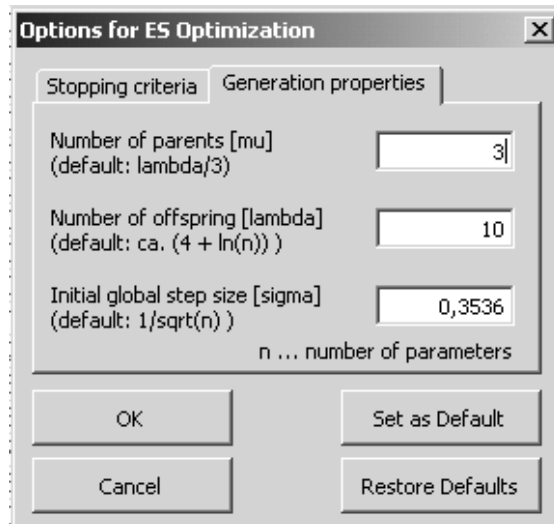


Abbildung 10: Erweiterte Einstellungen „Generation Properties“

- OptimizeWithESCMA
 - ParameterRange
 - ValueToOptimizeRange

Um auf die Subroutinen von einem anderen Projekt aus Zugriff zu haben, muss man zunächst einen Verweis auf das Addin machen (Extras ↦ Verweise).

5 Kurzanleitung

Aufgabe ist es an gegebene Messwerte ein Modellfunktion anzupassen. Die Beispielmesswerte in dieser Kurzanleitung sind dieselben wie aus Abbildung 4. Die Kurve ähnelt der Gaussfunktion, weshalb diese als Modellfunktion gewählt wird. Nachdem alle Arbeitsschritte, die zur Verwendung des Makros notwendig sind durchgeführt wurden, sieht die Exceltabelle wie in Abbildung 11 aus. Die Kurzanleitung referenziert im folgenden die eingezeichneten Punkte aus dieser Abbildung. Es wird davon ausgegangen, dass die Messwerte noch nicht eingegeben sind und das Makro nicht fest als Addin installiert ist.

1. Neues Excelsheet öffnen.
2. Macro Sicherheitseinstellungen anpassen (mittlere Sicherheitsstufe)
3. escma.xla laden
4. Eintragen der X- (Punkt 1 in der Abbildung) und Y1- (Punkt 2 in der Abbildung) Werte der Messdaten in jeweils eine Spalte.

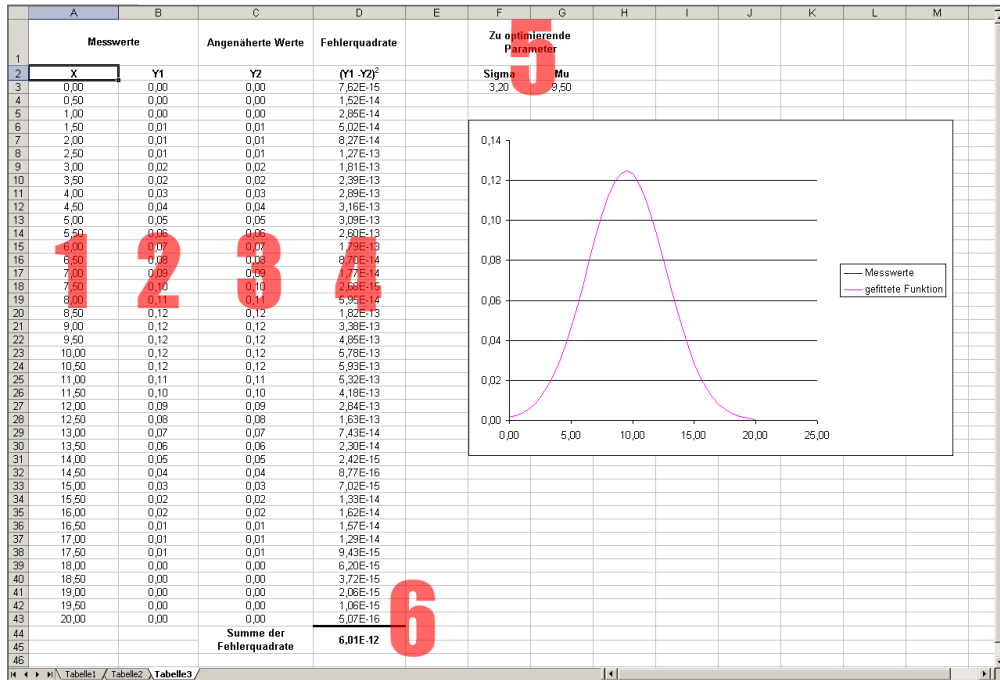


Abbildung 11: Screenshot einer Beispieldatei in Excel zur Verwendung des Addins

5. Eintragen der Parameter der zu approximierenden Funktion (Bsp. $ax^2 + bx + c$) (Punkt 5 in Abbildung 11) und auf einen Startwert setzen (z.B. 0 oder 1)
6. In einer neuen Spalte die Y2-Werte der zu approximierenden Funktion berechnen lassen. Die Y2-Werte an den Stellen auswerten, die auch durch die Messdaten vorliegen, also dieselben X-Werte für die zu fittende Funktion verwenden. (Punkt 3 in Abbildung 11)
7. Differenz zwischen errechneten (Punkt 3) und eingetragenen (Punkt 2) bilden und quadrieren (Punkt 4 in Abbildung 11)
8. Summe der Quadrierten Differenzen bilden (Punkt 6 in Abbildung 11)
9. Aus den Einträgen 1 und 2 eine Graphen zeichnen sowie aus 1 und 3 (beide in ein Diagramm).
10. Menü Extras \mapsto ES-Optimizer anwählen
11. Parameters to fit: Die Koeffizienten der Approximationsfunktion wählen (Punkt 5 in Abbildung 11)
12. Function to optimize: Die Summe der Fehlerquadrate auswählen (Punkt 6 in Abbildung 11)

13. Start drücken

Literatur

- [1] Rechenberg, Ingo: *Evolutionstrategie '94*, Stuttgart, Frommann-Holzboog, 1994
- [2] Hansen, Nikolaus: *Verallgemeinerte individuelle Schrittweitenregelung in der Evolutionstrategie : eine Untersuchung zur entstochastisierten, koordinatensystemunabhängigen Adaptation der Mutationsverteilung*, Mensch & Buch Verlag 1998