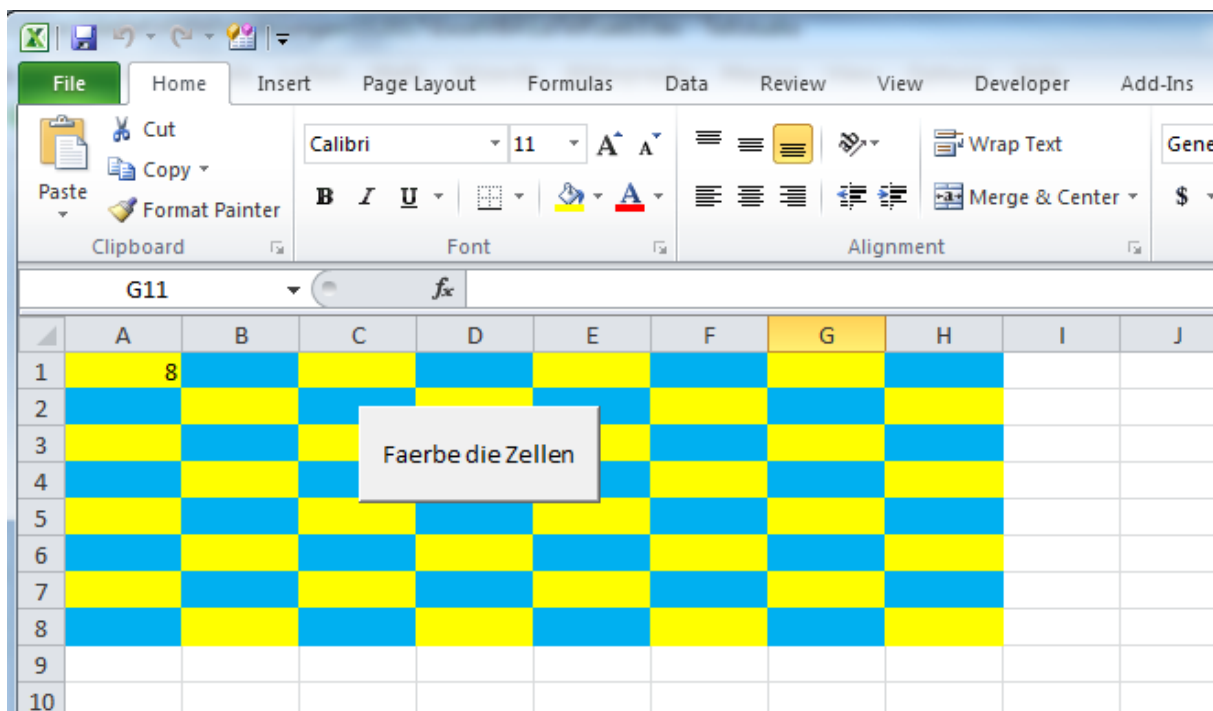


3. Übungsblatt zur Vorlesung Finanzmathematik mit Excel und VBA

Aufgabe 1) Lesen Sie sich in Urtis_ExcelVBA (bekommen Sie über den link <http://hsrc-mathematik.de/SS2017/semester4/ExcelVBA/book.pdf>) das Kapitel 2 'Getting Started with Macros' durch, das sind die Seiten 9-17. Insbesondere wird dort der Gebrauch des sogenannten Macro-Recorders erklärt. Mit dem Macro-Recorder können Sie jede Operation, die Sie auf einem Tabellenblatt ausführen, in VBA-code umsetzen und sich diesen code dann später anschauen. Der Macro-Recorder ist also ein sehr gutes Hilfsmittel wenn es darum geht, die genaue VBA-Syntax für eine bestimmte Operation (etwa: ich möchte eine Zelle gelb einfärben oder man möchte den Bereich A10:F110 markieren oder man möchte die Einträge in einer Spalte alphabetisch ordnen) herauszufinden.

Schreiben Sie dann ein Macro was folgendes macht: Aus der Zelle A1 wird eine natürliche Zahl n eingelesen und dann wird der Zellbereich von Cells(1,1) bis Cells(n , n) Schachbrettartig mit zwei verschiedenen Farben, etwa gelb und blau, eingefärbt. Eine eventuell schon vorhandene Einfärbung soll vorher eliminiert werden:



Aufgabe 2) Performance Excel-Funktionen vs. VBA-Funktionen: Wir wollen die Schnelligkeit der VBA-Funktion `log()` mit der Schnelligkeit der Excel Worksheet-Funktion

`Application.WorksheetFunction.Ln()` vergleichen. Dazu berechnen wir numerisch das Integral

$$\int_0^x \log(y) dy = x \log(x) - x \quad (1)$$

mit Hilfe einer Riemannschen Summe,

$$\int_0^x \log(y) dy \approx \sum_{i=1}^{N_x} \log(x_i) \Delta x \quad (2)$$

mit $x_i = i \Delta x$ und $N_x = x/\Delta x$ so dass $x = N_x \Delta x$ und wir betrachten nur solche x so dass N_x tatsächlich eine ganze Zahl ist.

Schreiben Sie ein VBA-Programm, welches nacheinander die folgenden 4 Loops ausführt:

(i) Der erste Loop soll die folgenden x-Werte

$$x \in \{0.01, 0.02, 0.03, \dots, 4.97, 4.98, 4.99, 5.00\} \quad (3)$$

in die Spalte A schreiben (die Schrittweite ist hier also $0.01 = 10^{-2}$).

(ii) Der zweite Loop soll die Riemannsche Summe auf der rechten Seite von (2) für die x-Werte aus (3) berechnen und in die Spalte B schreiben. Dazu soll

$$\Delta x = 10^{-6}$$

fest gewählt werden und der Logarithmus soll mit Hilfe der **VBA-Funktion** `log()` evaluiert werden.

(iii) Der dritte Loop soll die Riemannsche Summe auf der rechten Seite von (2) für die x-Werte aus (3) berechnen und in die Spalte C schreiben. Dazu soll

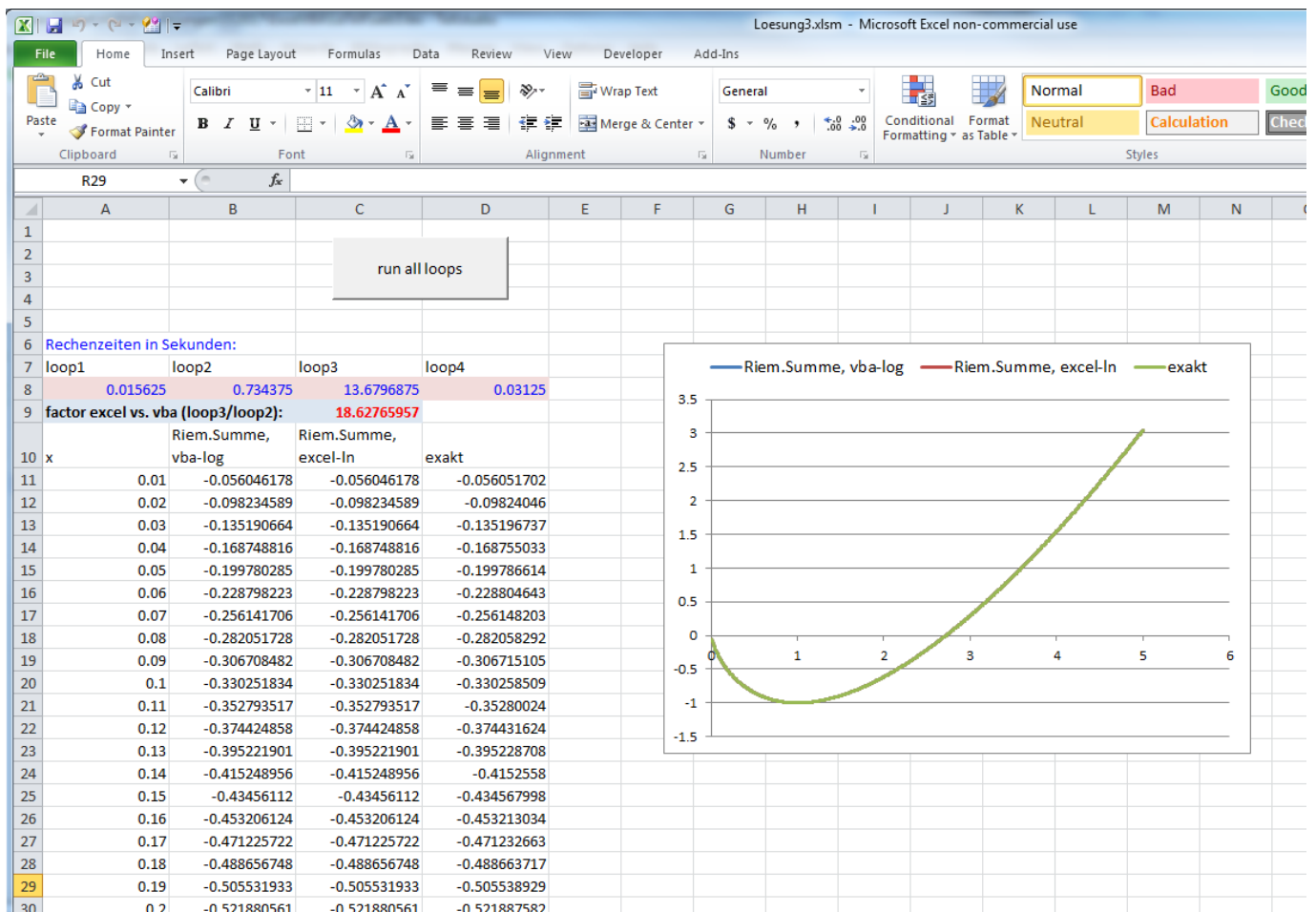
$$\Delta x = 10^{-6}$$

fest gewählt werden und der Logarithmus soll mit Hilfe der **Excel-Funktion** `LN()` evaluiert werden.

(iv) Der vierte Loop soll das exakte Resultat auf der rechten Seite von (1) für die x-Werte aus (3) berechnen und in die Spalte D schreiben. Dazu soll der Logarithmus mit Hilfe der VBA-Funktion `log()` evaluiert werden.

Es soll die Rechenzeit für jeden dieser 4 Loops berechnet werden und auf dem Excel-sheet ausgegeben werden. Ihre Ausgabe könnte also in etwa folgendermaßen aussehen:

..siehe nächste Seite



Zur Berechnung der Rechenzeiten der einzelnen Loops könnten Sie etwa die VBA Timer-Funktion zu Hilfe nehmen.

Hinweis: Da wir für das Δx in (2) den Wert 10^{-6} wählen, müssen wir, wenn wir die Variable $x_i = i\Delta x$ angelegt haben, die loop-Variable i von 0 bis 5'000'000 laufen lassen wenn wir auf $x = 5$ kommen wollen. Allerdings soll nicht die Riemannsche Summe für diese 5 Millionen x_i 's auf das sheet geschrieben werden, sondern eben nur für die x aus (3). Sie müssen sich also überlegen, für welche i 's man die x aus (3) 'trifft'.