Ludwig-Maximilians-Universität
München

Fortgeschrittenenpraktikum II
Wintersemester 22/23

Rheologie

Guido Osterwinter und Jan-Philipp Christ

München, den 17. Dezember 2022

Inhaltsverzeichnis

1. Zielsetzung und Motivation

Die Rheologie befasst sich mit dem Fließen und der Verformung von Materie, also mit dem Verhalten von Flüssigkeiten und Feststoffen unter dem Einfluss äußerer Kräfte. Die Viskoelastizität ist ein Teilgebiet der Rheologie, das die mechanischen Eigenschaften von Materialien untersucht, die sich wie eine Kombination aus einer viskosen Flüssigkeit und einem elastischen Festkörper verhalten. Viskoelastische Materialien zeigen sowohl viskoses als auch elastisches Verhalten als Reaktion auf Scherung oder andere mechanische Belastungen und sind beispielsweise in medizinischen oder industriellen Anwendungen von Relevanz.
Konkret sollen im Rahmen des hier vorgestellten Versuchs die viskoelastischen Eigenschaften von wässrigen Saccharose-Lösungen (Zuckerwasser) und von wässrigen Guaran-Lösungen verschiedener Konzentrationen untersucht werden. Für die genanten Lösungen werden unter Verwendung eines Rotationsrheometers die Scherrate und der Scherstress in Abhängigkeit der Scherrate vermessen.
Zuletzt wird eine 0.5-prozentige Wasser-Guaran-Lösung einer oszillatorischen Scherverformung, wobei die Oszillationsfrequenz bei fester Amplitude variiert wird, ausgesetzt, um damit die Abhängigkeit der viskoelastischen Materialantwort von der Frequenz zu beleuchten.

2. Theoretischer Hintergrund

Sofern nicht anders angegeben, stützen sich die Darstellungen in diesem Abschnitt auf [1]. Dieser wurden ebenfalls die gezeigten Bilder entnommen.

2.1. Elastizität und Viskosität

Ein Material heißt elastisch, wenn es, nachdem eine äußere Kraft angelegt wurde, eigenständig wieder in seine Ursprungsform zurückkehrt. Die gegensätzliche Eigenschaft zur Elastizität ist die Viskostiät. Ein Materialhat viskose Eigenschaften, wenn es, nachdem eine äußere Kraft angelegt wurde, in der Form bleibt, die es durch die äußere Kraft angenommen hat.
Aus dem Alltag kennen wir viele Stoffe, welche sich weder ausschließlich elastisch noch ausschließlich viskos verhalten, sondern sowohl elastische als auch viskose Eigenschaften haben. Solche Stoffe bezeichnet man als viskoelastisch.
Bei den elastischen Materialien gibt es eine wichtige Untergruppe: solche mit linear elastischem Verhalten. Ein Material heißt linear elastisch, wenn sich die Verformung beim Anlegen einer äußeren Kraft proportional zu der äußeren Kraft ändert, und sich die Verformung nicht ändert solange die äußere Kraft konstant bleibt. Die Verformungen welche man hierbei typischerweise betrachtet sind die relative Dehnung $\Delta L/L$ und der Scherwinkel $\gamma \approx \Delta L/h$.

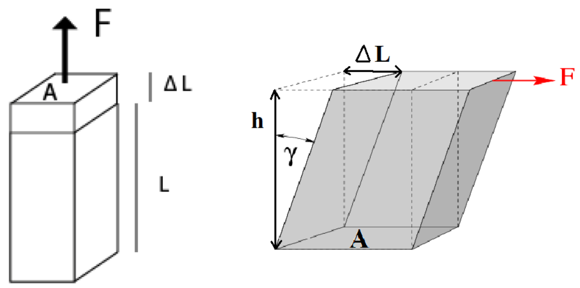


Abbildung 1 Skizze zur Dehnung und Scherung

Die Proportionalitätskonstanten, welche die Verformungen mit den entsprechenden Kräften verbinden sind, das elastische Modul E, wobei

$$\frac{F}{A} = E\frac{\Delta L}{L} \tag{1}$$

und das Schermodul G, wobei

$$\frac{F}{A} = G\frac{\Delta L}{h} \tag{2}$$

Es gibt allerdings auch Materialien bei denen sich die Verformungen nicht proportional zu den äußeren angelegten Kräften verhalten, wie z.B. Gummis oder andere Stoffe, die aus langen Polymermolekühlen bestehen. Aus der Alltagserfahrung heraus wissen wir auch, dass sich einige Stoffe viskoser, d.h. dickflüssiger verhalten als andere Stoffe. Aber wie kann man dieses Verhalten quantifizieren? Um die Viskosität eines Stoffes zu quantifizieren, wird beispielhaft folgender Aufbau benutzt:

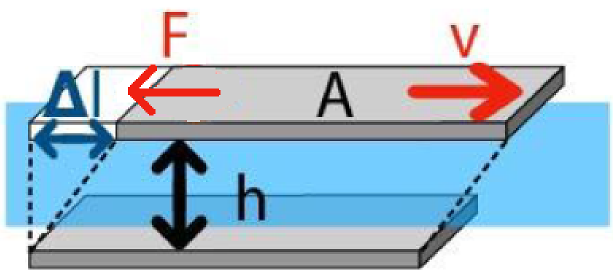


Abbildung 2 Experimentelle Viskositätsbestimmung

Hierbei füllt ein Stoff den Raum zwischen zwei parallelen Platten mit der Oberfläche A aus. Bewegt sich die obere Platte nun mit der Geschwindigkeit v über den Stoff, wird der Stoff eine Bremskraft $F(v)$ auf die Platte ausüben. Anschaulich ist klar, dass für die Bremskraft $F(v)$ gelten wird $F(v) \sim v \cdot A/h$. Die Viskosität η ist dann als

Proportionalitätskonstante definiert über die Gleichung

$$F(v) \;=\; \eta(v) \cdot v \cdot \frac{A}{h} \tag{3}$$

Mit den Größen $\dot{\gamma} \approx v/h$ (Scherrate) und $\sigma(v) = F(v)/A$ (Scherstress), kann man die Gleichung vereinfachen zu

$$\sigma(\dot{\gamma}) \;=\; \eta(\dot{\gamma}) \cdot \dot{\gamma} \tag{4}$$

Die Viskosität η ist eine Größe, welche, wie wir im Verlauf der beiden Experimente noch bestätigen werden, allgemein von der Scherrate $\dot{\gamma}$ abhängig ist.
Betrachtet man, wie hier in den Experimenten, Lösungen von bestimmten Stoffen mit Wasser, so ist die Viskosität einer Lösung auch konzentrationsabhängig. Für viele Stoffe liegt hier ein linearer Zusammenhang $\eta = \eta_0 + [\eta]c$ vor, wobei η_0 die Viskosität von Wasser ist, und $[\eta]$ die sogenannte intrinsische Viskosität des gelösten Stoffs. Für Stoffe aus langen Polymermolekühlen gilt diese Beziehung i.A. nicht. Hier nimmt die Viskosität in Abhängigkeit der Konzentration allgemein folgenden Verlauf an.

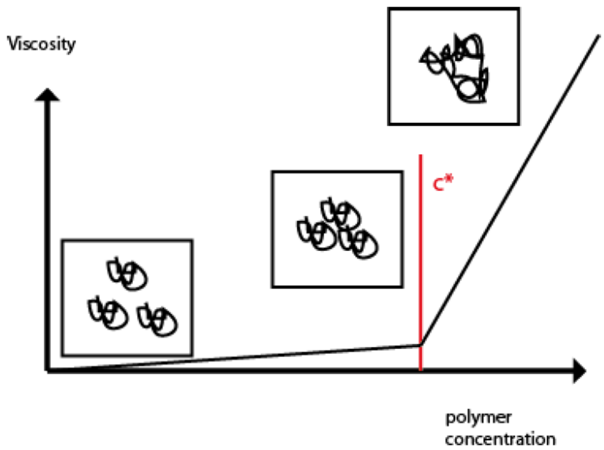


Abbildung 3 Polymerlösungen

Die Konzentration, bei der der charakteristische Knick auftritt, ist die sogenannte kritische Konzentration oder Überlappkonzentration c^*. Ab hier beginnen sich die Polymere aufgrund ihrer Größe mehr und mehr zu überlappen und ineinander zu verschlaufen, wodurch die Viskosität stark mit der Konzentration ansteigt.

2.2. Klassifizierung von Flüssigkeiten anhand ihres Fließverhaltens

Abhängig von der molekularen Strucktur einer Flüssigkeit kann die Viskosität in unterschiedlicher Art und Weise von der Scherrate abhängen. Man unterscheidet im wesentlichen drei Fälle.

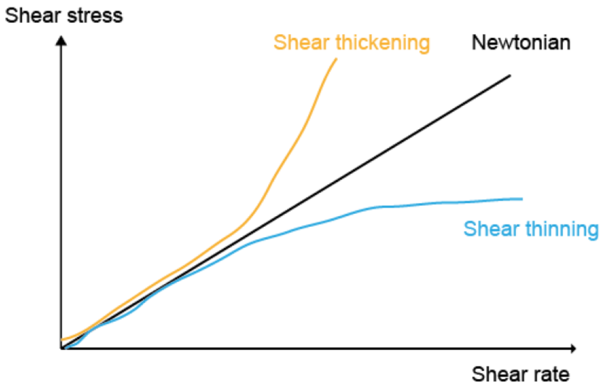


Abbildung 4 Unterschiedliche Fließverhalten im $\sigma - \dot{\gamma}$-Diagramm

- Newtonsches Fließverhalten:

  Newtonsche Flüssigkeiten zeichnen sich durch eine konstante Viskosität in einem großen Scherratenbereich aus. Der Scherstress ist daher bei diesen Flüssigkeiten direkt proportional zur Scherrate. Beispiele sind Wasser, Öl, Glycerol oder Honig.

- Scherverdünnung:

  Von Scherverdünnung spricht man, wenn die Viskosität mit der Scherrate abnimmt. Sie tritt auf, wenn die molekulare Strucktur der Probe aufgrund hydrodynamischer Kräfte, die durch die Scherung erzeugt werden, zerstört wird. Daher führt eine Erhöhung der Scherrate nur zu einer unterproportionalen Erhöhung des Scherstresses. Beispiele sind Duschgel, Joghurt, Zahnpasta, Marmelade.

- Scherverdickung:

  Von Scherverdickung spricht man wenn die Viskosität mit der Scherrate zunimmt. Die Ursache für Scherverdickung ist eine Volumenzunahme, oder eine Zunahme der Anzahl der struckturellen Untereinheiten der Probe, aufgrund der Scherung. Daher steigt der Scherstress überproportinal mit der Scherrate an. Beispiel: Stärkelösung.

Die Abhängigkeit des Scherstresses von der Scherrate kann nach dem Fließgesetz nach Ostwald und de Waele ($\rightarrow$ [2]) durch ein Potenzgesetz der Form

$$\sigma(\dot{\gamma}) = \beta \dot{\gamma}^{\alpha} \tag{5}$$

beschrieben werden. Für $\alpha = 1$ liegt Newtonsches Fließverhalten vor, für $\alpha < 1$ Scherverdünnung und für $\alpha > 1$ Scherverdickung.

2.3. Viskoelastizität

Zeigt ein Material sowohl elastisches und viskoses Verhalten, so kann dessen Antwort für hinreichend kleine Scherungen/ Deformationen/ Dehnungen als linear angenommen

werden. Dies führt für rein elastische Materialien zum Modell des Hookschen Körpers und für rein viskose Flüssigkeiten zu dem Modell der newtonschen Flüssgkeiten, die als Netzwerk von kleinen, bei Auslenkung Energie dissipierenden, Kolben beschrieben werden können. Das Maxwell-Modell kombiniert diese beiden Modelle, indem Materialien mit sowohl viskosen als auch elastischen Eigenschaften als Netzwerke von in Serie geschalteten Federn und Kolben interpretiert werden.
Legt man an solche Körper eine oszillatorische Verformung

$$\gamma(t) = \mathrm{Re}\left(\gamma_0 e^{i\omega t}\right) \tag{6}$$

an, so teilt sich die Stressantwort des Materials nach dem Maxwell-Modell in einen Term proportional zur Scherung und in einen Term proportional zur Scherrate auf, sodass

$$\sigma(t) = \mathrm{Re}\left(G\gamma(t) + \eta\dot{\gamma}(t)\right) = \mathrm{Re}\left(G\gamma_0 + \gamma_0\eta \cdot i\omega e^{i\omega t}\right) =: \mathrm{Re}\left(\sigma_0 e^{i\omega t+\delta}\right) \tag{7}$$

$$= \frac{\sigma_0}{\gamma_0}\mathrm{Re}\left(\gamma(t)(\cos\delta + i \cdot \sin\delta)\right) =: \mathrm{Re}\left((G' + iG') \cdot \gamma(t)\right) =: \mathrm{Re}\left(G^* \cdot \gamma(t)\right) \tag{8}$$

gilt mit dem Verlustmodul $G'' = \eta\omega$ und dem Speichermodul $G' = G$. Die Phase von G^* definiert den Verlustfaktor $\tan\delta \equiv G''/G'$.
Das Verlust- und das Speichermodul sind beide i.A. abhängig von der Art der Verformung. Legt man beispielsweise eine oszillatorische Verschiebung variabler Frequenz mit fester Amplitude an, so ergibt sich für Polymere in Wasserlösung typischerweise das folgende viskoelastische Spektrum:

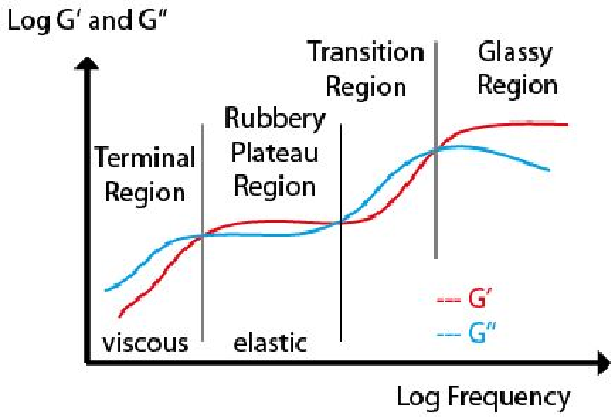


Abbildung 5 Viskoelastisches Spektrum einer Polymerlösung (aus [3])

Für kleine Frequenzen ist $G'' > G'$, wobei beide in guter Näherung linear mit der Frequenz zunehmen. In diesem Bereich hat der Stoff den Charakter einer viskoelastischen Flüssigkeit. Nachdem sich Speicher- und Verlustmodul gekreuzt haben, sind über einen gewissen Frequenzbereich (die Rubbery Plateau Region) die viskoelastischen Stofeigenschaften weitestgehend unabhängig von der Frequenz, und der Stoff verhält sich gelartig oder wie ein viskoelastischer Festkörper. In der sich daran anschließenden Übergangsregion dominieren die viskosen Eigenschaften des Körpers. Für höhere Frequenzen wird die Glassy Region erreicht, in der das Speichermodul plateauartig wird, wohingegen das Speichermodul abnimmt. Der Stoff verhält sich also zunehmend wie ein (elastischer) Festkörper.

2.4. Rotationsrheometer

Im Versuch wurde das Rotationsrheometer KINEXUS ultra+ von Malvern Panalytical verwendet. Die Probe wird zwischen zwei runden Metallplatten aufgetragen, von denen die obere drehbar gelagert ist und mit variabler Winkelgeschwindigkeit Ω rotieren kann. Der Plattenabstand ist variabel, wird aber bei den Messungen hier fest auf $d = 1$ mm gesetzt.
Bei einer viskosen Probe wird durch das Rotieren eine Stressantwort $\sigma \propto \eta$ resultieren, welche proportional zum Drehmoment M ist, welches auf die obere Platte wirkt. Dieses Drehmoment wird mit einer Auflösung von 0.05 nNm gemessen (vgl. [4]), was bei Kenntnis der Normalkraft, die auf die Platte wirkt, und einiger geometrieabhängiger Kenngrößen das Berechnen der Stressantwort und damit der Viskosität zulässt.
Wegen $\eta \propto M$ ist $\frac{\Delta\eta}{\eta} = \frac{\Delta M}{M}$ und damit der relative Fehler bei der Viskositätsbestimmung für weniger viskose Flüssigkeiten größer.

3. Versuchsdurchführung

3.1. Wasser-Saccharose

Im ersten Experiment wurde die Viskosität in Abhängigkeit der Scherrate für unterschiedlich konzentrierte Saccharose-Wasserlösungen gemessen. Die untersuchten Konzentrationen waren $0\%, 10\%, 20\%$ und 40%. Zum Messen der Viskositäten in Abhängigkeit der Scherraten wurde das in Abschnitt 2.4 vorgestellte Rheometer zusammen mit dem Programm 'Shear Rate Table for Waterlike Samples' benutzt. Der Scherratenbereich, in welchem gemessen wurde, ging von 10^0 s^{-1} bis 10^2 s^{-1} mit 5 Messpunkten pro Dekade. Tskip wurde auf 1 min eingestellt. Zum Vermessen der Lösungen wurde jeweils immer etwas mehr Lösung angemischt als das verwendete Rheometer eigentlich gebraucht hätte. Die genau verwendeten Mengen Wasser und Saccharose für die jeweiligen Lösungen können Tabelle 2 entnommen werden.
Beim Anmischen der Lösungen wurde stets darauf geachtet, dass die verwendete Saccharose komplett gelöst war. Die Temperatur, bei welcher die verschiedenen Lösungen vermessen werden sollten, wurde im Bedienprogramm des Rheometers auf $25°C$ eingestellt. Das Profil der Proben zwischen den beiden Platten des Rheometers hatte jeweils immer in etwa folgende Gestalt:

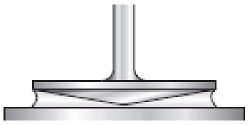


Abbildung 6 Flüssigkeitsprofil bei den Saccharose-Wasserlösungen (aus [3])

Nachdem eine Messreihe fertig war, wurden jeweils die beiden Platten des Rheometers mit einem Tuch und Reinigungsalkohol gereinigt.

3.2. Wasser-Guaran

3.2.1. Anmischen der Lösungen

Prinzipiell gleicht das Vorgehen beim Anmischen und Auftragen der Lösungen auf das Rheometer dem in Abschnitt 3.1. Da Guaran bei Wasserkontakt schnell eine viskose Flüssigkeit bildet, die nur mit großen Schwierigkeiten pipettiert werden kann, wurde die jeweilige Probe unmittelbar vor Messbeginn mit Wasser versetzt. Dabei war auf ein möglichst schnelles Vermischen von Guaran und Wasser zu achten, da Guaran die Tendenz hat, bei Wasserkontakt Klumpen auszubilden, die ein vollständiges Durchmischen der gesamten Pulversubstanz erheblich erschweren. Eine solche Klumpenbildung war bei der 1%igen-Lösung zu beobachten. Durch mehrfaches Pipettieren konnten die Klumpen so weit aufgelöst werden, dass keine mehr mit bloßem Auge zu erkennen waren. Im Prozess bildeten sich in der Flüssigkeit jedoch kleine Luftblasen, die nicht vollständig von der Flüssigkeit getrennt werden konnten. Auf deren Einfluss auf die Messergebnisse wird in Abschnitt 4.2.3 eingegangen.
Eine weitere Schwierigkeit beim Anmischen der Lösungen ist, dass gerade bei höheren Konzentrationen ein nicht unerheblicher Anteil der Substanz wegen der hohen Viskosität nach dem Einziehen in der Pipette verbleibt. In besonderem Maße wurde deshalb beim Loaden des Samples darauf geachtet, dass eine ausreichende Abdeckung des Messstempels am Rheometer erzielt wurde, um Underfilling zu vermeiden. So musste beispielsweise für die 2.3%ige Lösung nachträglich noch Flüssigkeit hinzugefügt werden, da anfangs nicht das gesamte Volumen unter der Geometry-Platte von Flüssigkeit ausgefüllt wurde. Auf diese Weise wurde ein Zustand erreicht, bei dem die Flüssigkeit sich so aus dem Spalt herauskrümmte, dass der Krümmungsradius unterhalb der Geometry lag.

3.2.2. Scherratenmessungen

Die sich anschließende Messung wurde nach Vornehmen der entsprechenden Einstellungen unter Toolkit_V001 Shear Rate Table mit $\dot{\gamma} \in [1, 100]\mathrm{s}^{-1}$, $T = 25°C$, 5 Messpunkten pro Dekade und dem Parameter Tskip nach 1 min vom Rheometer automatisiert durchgeführt, woraufhin die Messwerte als Tabelle exportiert werden konnten.
Es wurden Konzentrationen $c \in [0, 0.25, 0.5, 1.0, 1.4, 2.0, 2.3]\%$ untersucht, wobei die

Messwerte für die reine Wasserlösung mit $c = 0.0\%$ aus der vorangegangenen Saccharose-Messreihe übernommen werden konnten.

3.2.3. Frequenzversuch

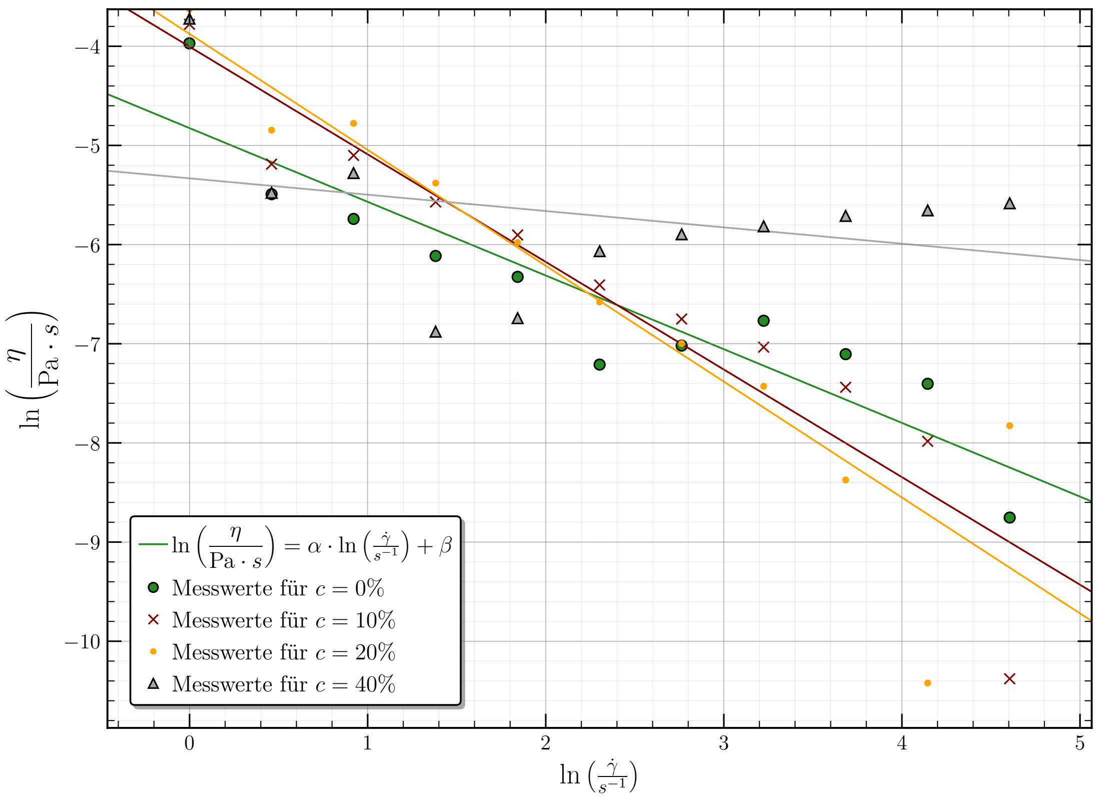
In diesem Versuchsteil sollte eine 0.5%ige Guaran-Lösung einer oszillatorischen Scherverformung ausgesetzt werden. Es wurde dieselbe Probe, mit der auch die entsprechende Messung in Abschnitt 4.2.1 durchgeführt wurde, verwendet. Der eigentliche Messprozess lief automatisiert nach Aufrufen des entsprechenden Programms Toolkit_O002 Frequency Table in der Rheometer-Software und dem Einstellen des Messbereichs von 0.01 − 200 Hz, der Soll-Temperatur $T = 25°C$ und der Anzahl (5) von Messpunkten pro Dekade. Tskip wurde auf 1 min gesetzt, die Stärke der Scherverformung betrug 0.5%.
Aufgezeichnet wurden das Speichermodul G' und das Verlustmodul G'' als Funktion der Oszillationsfrequenz bei konstanter Amplitude der Scherverformung.

4. Ergebnisse und Diskussion

4.1. Wasser-Saccharose

4.1.1. Scherratenmessungen

Die Messreihen der Viskosität in Abhängigkeit der Scherrate für die verschiedenen Saccharose-Wasser-Lösungen sowie die zugehörigen Fitgeraden sind in folgendem Diagramm dargestellt:



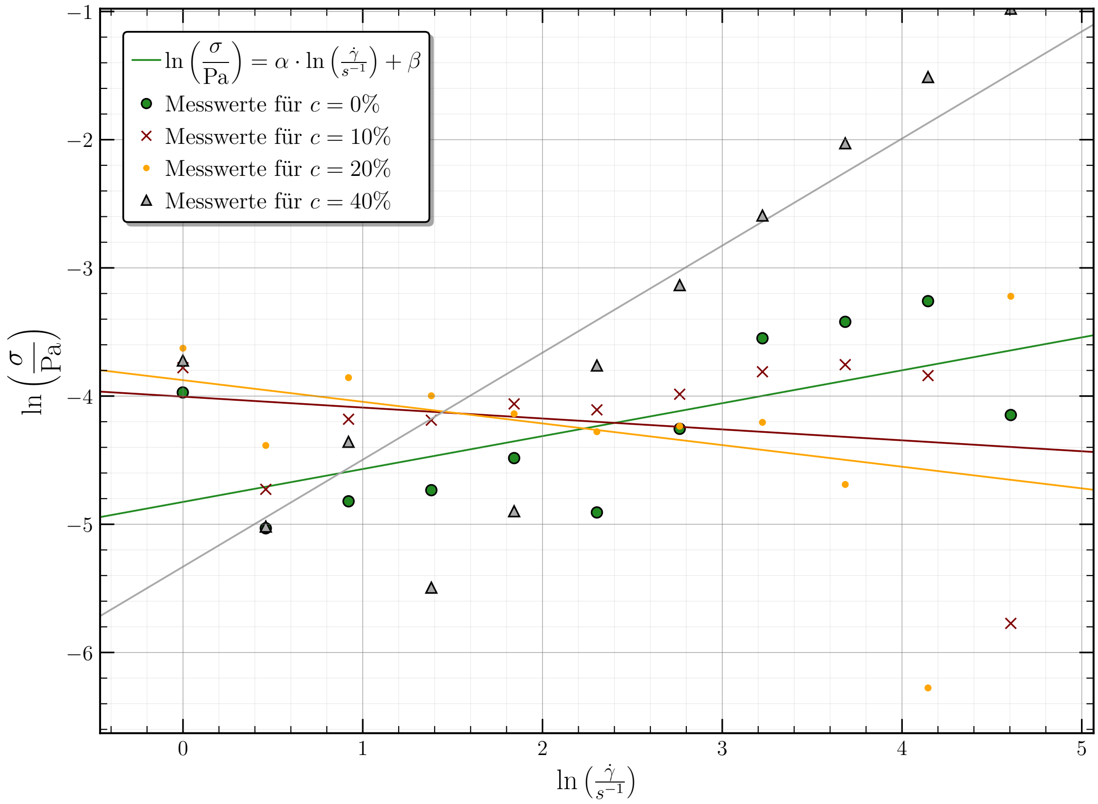
	$c = 0\%$	$c = 10\%$	$c = 20\%$	$c = 40\%$
α	-0.743 ± 0.11	-1.085 ± 0.13	-1.169 ± 0.16	-0.165 ± 0.17
β	-4.825 ± 0.29	-4.005 ± 0.34	-3.876 ± 0.43	-5.332 ± 0.47

Plot 1 Auftragung von η gegen $\dot{\gamma}$ bei Saccharose

Aus den Steigungen und Ordinatenverschiebungen der Fitgeraden kann man gemäß

$$\eta = e^{\beta}\gamma^{\alpha} \iff \ln(\eta) = \alpha \cdot \ln(\dot{\gamma}) + \beta \tag{9}$$

für die einzelnen Lösungen die Potenzgesetzbeziehung zwischen Scherrate und Viskosität bestimmen. Im nachfolgenden Diagramm sind für die verschiedenen Lösungen, die sich aus den Messwerten ergebenden Scherspannungen σ gegen die Scherrate $\dot{\gamma}$ aufgetragen, wodurch gegenüber Plot 1 wegen der Beziehung $\sigma = \eta\dot{\gamma}$ keinerlei neue Information gewonnen wird:

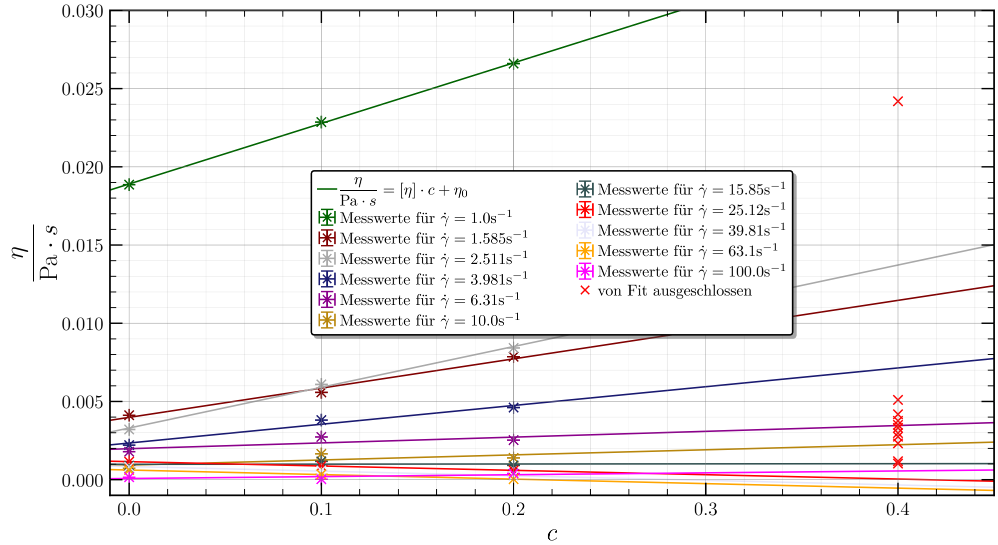


	$c = 0\%$	$c = 10\%$	$c = 20\%$	$c = 40\%$
α	0.257 ± 0.11	-0.085 ± 0.13	-0.169 ± 0.16	0.835 ± 0.17
β	-4.826 ± 0.29	-4.004 ± 0.34	-3.876 ± 0.43	-5.331 ± 0.47

Plot 2 Auftragung von σ gegen $\dot{\gamma}$ bei Saccharose

Da die Steigungen aller vier Fitgeraden < 1 sind, deuten die Messwerte für alle vier Lösungen qualitativ auf Scherverdünnung hin. Da jedoch $[\alpha - \Delta\alpha, \alpha + \Delta\alpha] = [0.835 - 0.17, 0.835 + 0.17] \cap [1, +\infty) \neq \emptyset$, kann prinzipiell die These, die 40%ige Lösung verhalte sich newtonsch oder scherverdickend, nicht verworfen werden, wenngleich sie unwahrscheinlich erscheint. Auch muss für alle Messreihen die starke Streuung der Messwerte angemerkt werden, was es erschwert, jeweils einen klar linearen Trend auszumachen. Dies äußert sich in den großen relativen Unsicherheiten der Fitparameter α, β bei allen Konzentrationen.
Im nächsten Diagramm sind für die verschiedenen Scherraten, für welche die vier Messreihen aufgenommen wurden, die Viskositäten in Abhängigkeit der Konzentrationen der

Lösungen dargestellt:



	$\dot{\gamma} = 1.000\mathrm{s}^{-1}$	$\dot{\gamma} = 1.585\mathrm{s}^{-1}$	$\dot{\gamma} = 2.511\mathrm{s}^{-1}$	$\dot{\gamma} = 3.981\mathrm{s}^{-1}$	$\dot{\gamma} = 6.310\mathrm{s}^{-1}$	$\dot{\gamma} = 10.000\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$\frac{[\eta]}{\mathrm{mPa}}$	387.0 ± 7.5	187.0 ± 24	260.5 ± 16	120.0 ± 23	37.0 ± 33	32.55 ± 34	0.95 ± 15	-27.8 ± 0.52	-29.5 ± 3.6	-28.96 ± 1.2	12.05 ± 14
$\frac{\eta_0}{\mathrm{mPa}}$	189.033 ± 0.97	39.83 ± 3.1	33.02 ± 2	23.43 ± 3	19.8 ± 4.2	9.34 ± 4.4	9.84 ± 2	11.53 ± 0.067	8.417 ± 0.46	6.162 ± 0.16	0.76 ± 1.8

Plot 3 Auftragung von η gegen c bei Saccharose

Beim Erstellen der Fitgeraden für die verschiedenen Scherraten wurden hier allerdings die Messwerte der 40%igen Saccharoselösung ausgeschlossen, da diese dem linearen Trend, der klar von den anderen drei Lösungen vorgegeben wird, widerspricht. Auf den Umstand, warum ein anderes Verhalten bei den Messwerten zur 40%igen Saccharoselösung plausibel ist, wird später noch eingegangen.
Die sich aus den Fitgeraden ergebenden intrinsischen Viskositäten $[\eta]$ für Saccharose sind in der Tabelle in Plot 3 aufgeführt. Aufgrund der sehr schwankenden Werte ist es hier aber nicht sinnvoll für diese einen Mittelwert anzugeben, sodass nicht abschließend ein Wert für die intrinsische Viskosität von Saccharoselösungen angegeben werden kann.
Abgesehen von den Messwerten der 40%igen Saccharoselösung, die bei dem Fit in Plot 3 nicht berücksichtigt wurden, zeigen alle anderen Messwerte jeweils nur sehr geringe Abweichungen zu ihrer jeweiligen Fitgeraden. Dies ist konsistent mit dem gemäß Abschnitt 2.1 für Saccharoselösungen zu erwartenden linearen Zusammenhang zwischen Konzentration und Viskosität. Denn Saccharosemoleküle sind keine langen Kettenmolekühle und haben daher nur wenig Tendenz, sich zu verhaken, sodass für Saccharoselösung eine feste Steigerung oder Verringerung der Konzentration unabhängig von der Ausgangskonzentration die gleiche Änderung in der Viskosität verursacht. Für Saccharoselösungen ist zu erwarten, dass diese sich in einem Scherratenbereich von 1 s^{-1} bis 130 s^{-1} newtonsch verhalten ([5], S.37). Dies steht im Widerspruch zu den in Plot 1 dargestellten Mess-

werten. Lediglich für die 40%ige Lösung haben wir gemäß Plot 2 Viskositätsmesswerte erhalten, welche in etwa konstant sind. Das scherverdünnende Verhalten der 0, 10 und 20 prozentigen Lösungen, welches in Plot 1 und Plot 2 beobachtet werden kann, kann dadurch erklärt werden, dass die Lösungen die im Messprogramm eingestellte Soll-Temperatur von $25°C$ beim Start der Messreihen wahrscheinlich noch nicht erreicht hatten, und sich noch während der Messreihdurchführung weiter aufgewährmt haben. Dies erscheint besonders plausibel, da, wie Stadler in Ziffer 2.1.2 in [6] anmerkt, die vom Rheometer bestimmte Temperatur nicht *im* Sample gemessen wird, sondern in der Platte darunter oder darüber.

Die Lösungen bei den kleinen Scherraten waren also wahrscheinlich immer noch etwas kälter waren als bei den großen Scherraten, da das Rehometer die Viskositäten zuerst für die kleinen Scherraten gemessen hat. Und da die Viskosität eines Stoffs allgemein mit steigender Temperatur abnimmt, ergibt sich dadurch die scheinbare Scherverdünnung. Verstärkt wird dieser Effekt durch die geringe Umgebungstemperatur von $\sim 13°C$ im Versuchsraum, welche es unwahrscheinlich erscheinen lässt, dass sich das Sample in der Aufwärmphase an die im Rheometer eingestellte und gemessene Temperatur einstellen konnte.

Der Unterschied der drei Messreihen zu $c = 0\%, 10\%, 20\%$ zur Messreihe der 40%igen Lösung ist, dass der 40%igen Lösung mehr Zeit gegeben wurde sich vor Messbeginn auf die $25°C$ aufzuwärmen aufgrund des Umstands, dass diese Messung vor der Mittagspause gestartet wurde und der Aufwärmprozess deshalb voll durchlaufen konnte.

D.h. bei einer erneuten Versuchdurchführung ist genauer darauf zu achten, dass den Proben auch wirklich genug Zeit gegeben wird, ihre Soll-Temperaturen zu erreichen. Denkbar wäre es, die angemischten Lösungen zuvor in ein (temperiertes) Wasserbad zu legen, um sie vorzuheizen. Vor dem Loaden des Samples könnte die Temperatur mit einem Thermometer überprüft werden.

Vor diesem Hintergund sind die Viskositätsmesswerte für die hohen Scherraten wahrscheinlich auch die genauesten und weichen von Plot 3 von den Messwerten zu den übrigen Konzentrationen ab. Vergleicht man die Viskositätsmesswerte der hohen Scherraten für die verschiedenen Konzentrationen mit den tabellierten Literaturwerten aus [7] auf S.4, ergibt sich, dass die Literaturwerte größer sind als unsere gemessenen Werte:

$c[\%]$	$\eta[\mathrm{Pa}\cdot\mathrm{s}]$	
	Literatur	Messwerte für $\gamma > 20\ \mathrm{s}^{-1}$
10	1.13	0.91
20	1.7	0.75
40	5.17	3.7

Tabelle 1: Viskosität bei hohen Scherraten

Dies liegt wahrscheinlich am 'Underfilling', also dass wir aufgrund mangelnder Erfahrung die Proben, welche wir zwischen die Platten der Rheometers gegeben haben, stets etwas zu klein gewählt haben. Denn im Profil hatten unsere Proben immer etwa die in Abbildung 6 gezeigte Gestalt. Bei korrekter Größe der Proben hätte das Profil aber immer wie folgt aussehen sollen:

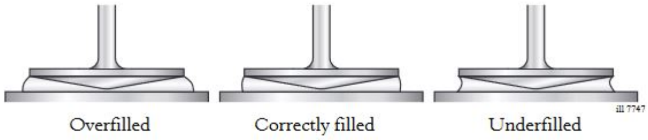


Abbildung 7 Zur korrekten Platzierung des Samples (aus [3])

Wie in [8] auf S.3 erwähnt, kann aber bereits eine 100 μm große Änderung des Radiuses r der Probe zu einem 1.6%igen Fehler in der gemessenen Viskosität führen, wobei das konkrete Ausmaß des Fehlers natürlich noch vom Soll-Radius der Probe abhängt. Die von Hellström angegebenen Werte sind hier nur genannt, um zu verdeutlichen, wie stark die Viskosität vom Sample-Radius abhängt.
Da eine kleinere Probe ein kleineres Rückstelldrehmoment erzeugt, welches das Rheometer misst, um daraus die Viskosität zu berechnen, führt eine kleinere Probe auch zu kleineren Viskositätsmesswerten. Bei einer erneuten Versuchdurchführung ist also insbesondere genauer darauf zu achten dass das Profil der Proben zwischen den beiden Rheometerplatten der Vorschrift in Abbildung 7 folgt.

4.1.2. Fehlerabschätzung zum Anmischen der Lösungen

Es ist interessant, abzuschätzen, ob mit den verwendeten Wasser- und Saccharosemengen tatsächlich die gewünschten Konzentrationen erreicht werden konnten, da beim Anmischen einige Vereinfachungen angenommen wurden: So wurde beim Abmessen der Wassermenge, die notwendig ist, um gegeben einer zu lösenden Stoffmenge eine bestimmte Konzentration zu erreichen, die Dichte von Wasser auf $\rho_W = 1\ \frac{\text{g}}{\text{ml}}$ gesetzt. Dies gilt bei Raumtemperatur und Normaldruck nicht exakt.
Mittels [9] kann die Dichte bei Raumtemperatur $T = 25°\text{C}$ und Normaldruck mit $\rho = (0.99705 \pm 0.0015)\ \frac{\text{g}}{\text{ml}}$ angegeben werden. Die Unsicherheit kommt dadurch, dass das Wasser erstens wahrscheinlich eine geringere Temperatur als die Luft im Labor hatte, da das Wasser auch über gewisse Zeiten hinweg im deutlich kälteren Versuchsraum mit dem Rheometer stand, und zweitens der Luftdruck nicht genau bekannt ist.
Ferner wurden die Messunsicherheiten der verwendeten Präzisionswaage und Pipette vernachlässigt. Die Unsicherheit in Messungen der Waage werden auf $\Delta m = 0.0002$ g und die Unsicherheit bei Messungen mit der Pipette auf $\Delta V = 2\mu$l geschätzt. Damit lassen sich die folgenden Werte für die relativen Unsicherheiten in den Stoffkonzentrationen gewinnen:

c[%]-Soll	$V_{H_2O}[\mu l]$	m[g]	$c = \frac{m}{\rho \cdot V + m}$[%]	Δc[%]	$\frac{\Delta c}{c}$[%]
10	1233	0.13736	10.0503	0.02394	0.2382
20	857	0.21426	20.0481	0.04695	0.2342
40	518	0.34569	40.0957	0.10049	0.2506

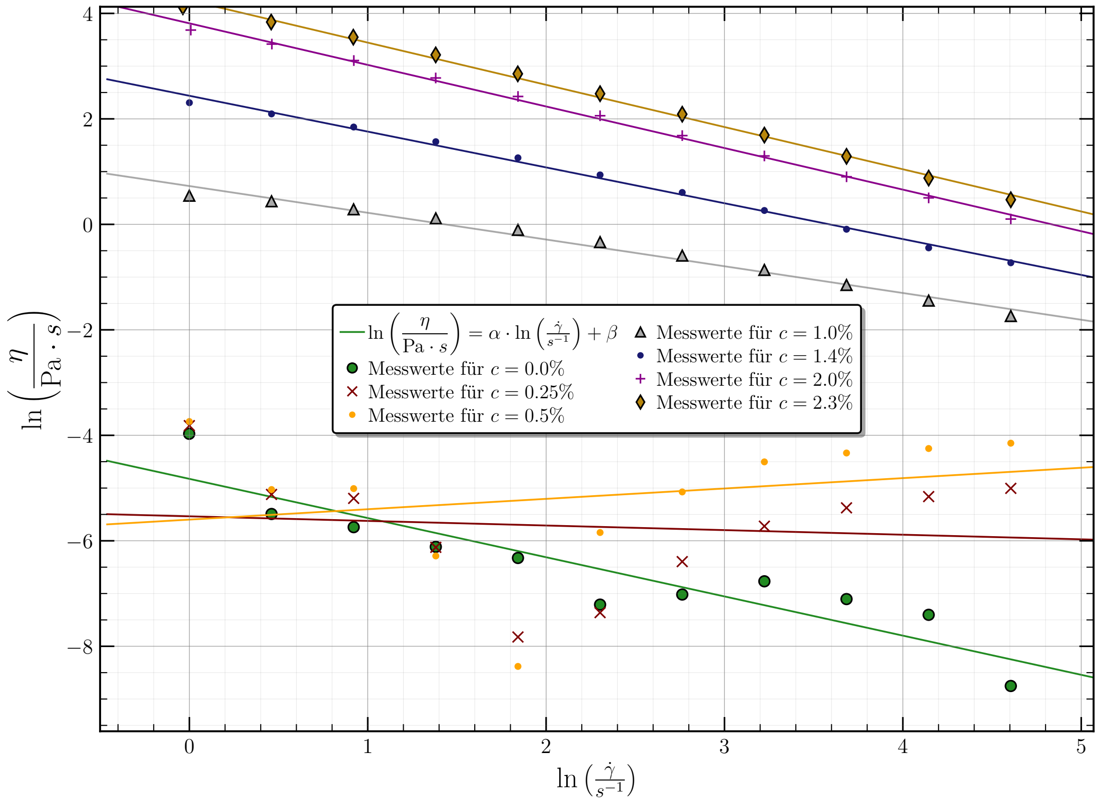
Tabelle 2: Saccharose-Konzentrationen

Die Konzentrationen hatten also im Rahmen der geringen Messunsicherheit in der Tat die Soll-Konzentration der jeweiligen Stoffe, sodass eine zu große oder zu kleine untersuchte Konzentration wegen ungenau abgemessener Massen oder Volumina bei den Saccharose-Wasser-Lösungen keine relevante Fehlerquelle ist und die in Tabelle 1 auftretende Diskrepanz nicht erklären kann.

4.2. Wasser-Guaran

4.2.1. Scherratenmessungen

Trägt man die für die verschiedenen Konzentrationen und Scherraten aufgenommenen Messwerte gegeneinander auf, so können die Werte $(\eta_i, \dot{\gamma}_i)$ für feste Konzentration c nach dem Fließgesetz von Ostwald und de Waele (vgl. [2]) in einem doppelt-logarithmischen Plot durch Geraden beschrieben werden.



	$c = 0.0\%$	$c = 0.25\%$	$c = 0.5\%$	$c = 1.0\%$	$c = 1.4\%$	$c = 2.0\%$	$c = 2.3\%$
α	-0.743 ± 0.11	-0.087 ± 0.25	0.197 ± 0.28	-0.507 ± 0.021	-0.6794 ± 0.014	-0.7883 ± 0.014	-0.8004 ± 0.017
β	-4.825 ± 0.29	-5.536 ± 0.67	-5.598 ± 0.76	0.7265 ± 0.059	2.4383 ± 0.039	3.8117 ± 0.037	4.2465 ± 0.045

Plot 4 Doppelt logarithmische Auftragung von η gegen $\dot{\gamma}$ bei Guaran

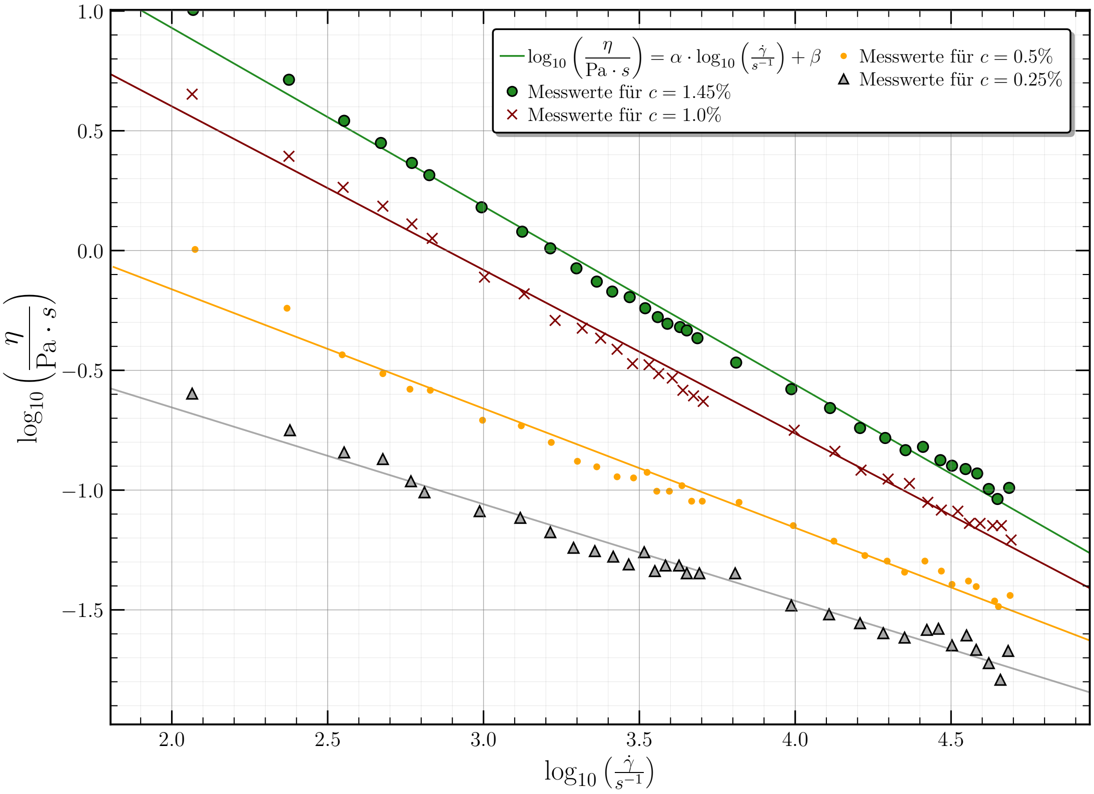
Rein qualitativ stellt man durch Inspektion von Plot 4 fest, dass die Messwerte für $c \in [1.0, 1.4, 2.0, 2.3]\%$ gut durch eine Gerade beschrieben werden können, was die Gültigkeit des Fließgesetzes von Ostwald und de Waele stützt und sich in einer geringen Standardabweichung der Fitparameter α („Steigung“) und β („Verschiebung entlang der Ordinate“) niederschlägt. In guter Näherung parallel zu den vier genannten Geraden verläuft die Fitgerade, die die Messwerte für die reine Wasserlösung beschreiben soll. Es

fällt die große Streuung der Messwerte auf, was auf die Bemerkung in 2.4, wonach der relative Fehler in der Viskosität bei weniger viskosen Flüssigkeiten größer sein sollte, zurückzuführen ist. Durch den größeren Fehler bei jedem Messwert wird die statistische Streuung um die Fitgerade und damit auch die Unsicherheit der Fitparameter α, β größer.
Unerwartet hingegen ist das Verhalten, das sich für $c = 0.25\%$ und $c = 0.5\%$ zeigt. Für die 0.5%ige Lösung legt der Fit gar ein $\alpha > 0$ nahe, was bedeuten würde, dass die untersuchte Lösung scher*verdickendes* Verhalten zeigen würde. Dies steht im Widerspruch dazu, dass Guaran nach [10] stark scher*verdünnende* Eigenschaften hat. Tatsächlich ist aber ein lineares Modell höchst unzureichend zur Beschreibung der betreffenden Daten. Denn nur für die letzten 4-5 Messwerte (bei hohen Scherraten) lässt sich ein linearer Zusammenhang (aber mit positiver Steigung!) erahnen. Für kleinere Scherraten sieht es vielmehr so aus, als ob die Daten zwei nach unten geöffneten Hyperbelästen folgen würden, die sich etwa bei $\ln\left(\frac{\dot{\gamma}}{\mathrm{s}^{-1}}\right) \approx 1.8$ treffen. Entsprechend schwach in der Aussagekraft sind die jeweiligen Fitparameter α, die je mit einer mehr als hundertprozentigen Unsicherheit behaftet sind. Insbesondere kann also anhand von Plot 4 nicht darauf geschlossen werden, dass Guar für manche Konzentrationen scherverdickendes Verhalten zeigt. Da der Mechanismus, der bestimmt, ob ein Stoff scherverdünnende oder -dickende Eigenschaften hat, prinzipiell für alle Konzentrationen derselbe ist (bei Polymeren wie Guaran: „Verhaken“ der langkettigen Moleküle in der Lösung), muss auch eine 0.5%ige Guar-Wasser-Lösung eindeutig scherverdünnende Eigenschaften haben und das erläuterte Verhalten bei $c = 0.25\%, 0.5\%$ auf systematische Messfehler zurückzuführen sein (siehe dazu Abschnitt 4.2.3).

Weiterhin muss angemerkt werden, dass Wasser als Standardbeispiel für ein Newtonsches Fluid eigentlich eine scherratenunabhängige Viskosität haben müsste.
Erklärungsansätze, warum dennoch scherverdünnendes Verhalten beobachtet wurde, wurden bereits in Abschnitt 4.1.1 gefunden.
Für $c = [1.0, 1.4, 2.0, 2.3]\%$ entspricht das beobachtete Verhalten der Erwartung, dass Guaran in Wasserlösung ein scherverdünnendes Fluid ist. Die Potenzabhängigkeit ist also von der Form

$$\eta \propto \dot{\gamma}^{\alpha} \tag{10}$$

mit $\alpha \in [-0.51 \pm 0.03, -0.68 \pm 0.02, -0.79 \pm 0.02, -0.80 \pm 0.02]$. Die Güte und Aussagekraft dieser Werte ist hoch; lediglich eine Veschiebung dieser Werte durch systematische Fehlereinflüsse ist denkbar. Diese werden in Abschnitt 4.2.3 eingeschätzt. Für die drei niedrigesten Konzentrationen hingegen ist aus den oben erläuterten Gründen ein abschließendes Festhalten des Fitparameters α zur Bestmmung der Potenzgesetz-Beziehung nicht sinnvoll.
In [11] wurden ebenfalls Messungen zur Scherratenabhängigkeit von der Viskosität vorgenommen. Abbildung 8 aus diesem Paper können Messdaten für Guaran-Wasser-Lösungen der Konzentrationen $0.25\%, 0.5\%, 1.0\%, 1.45\%$ entnommen werden. Diese Daten können wiederum logarithmisch aufgetragen und linear gefittet werden. Man erhält Plot 5 :

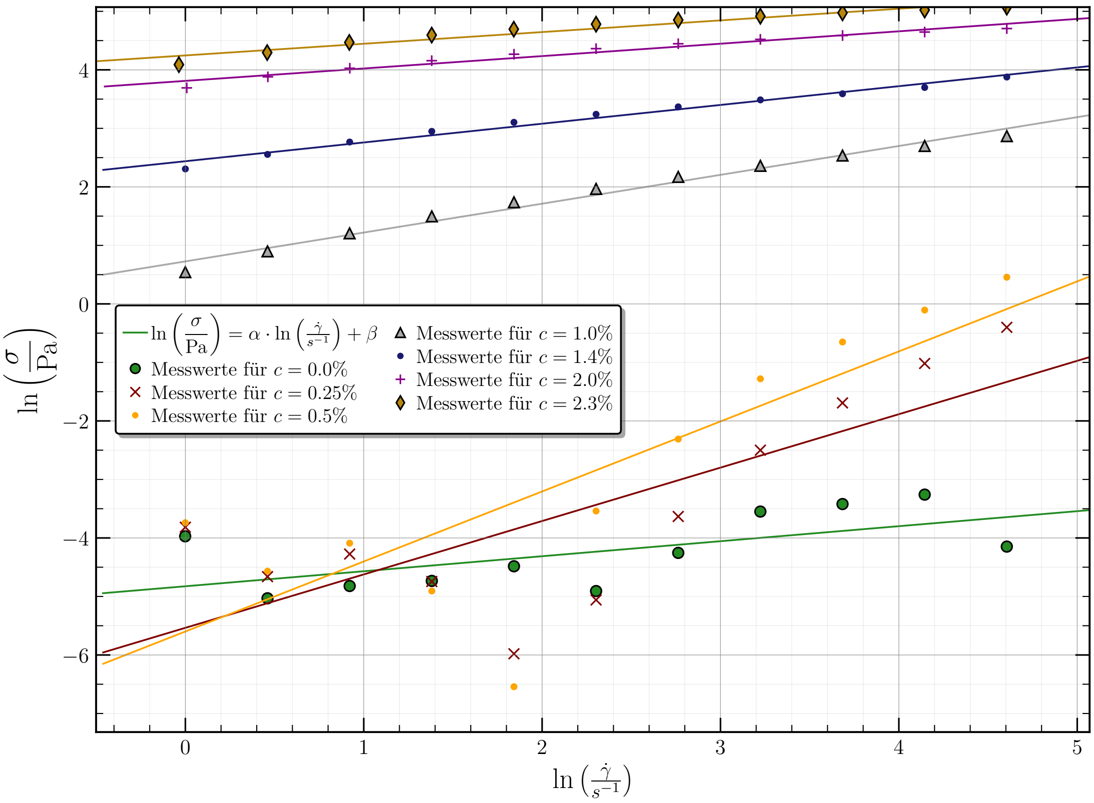


	$c = 1.45\%$	$c = 1.0\%$	$c = 0.5\%$	$c = 0.25\%$
α	-0.7443 ± 0.011	-0.683 ± 0.011	-0.4978 ± 0.014	-0.4038 ± 0.011
β	2.418 ± 0.04	1.9681 ± 0.039	0.834 ± 0.052	0.153 ± 0.04

Plot 5 Doppelt logarithmische Auftragung von η gegen $\dot\gamma$ mit Messwerten aus [11]

Zunächst fällt die qualitative Überlegenheit in der Güte dieser Daten insbesondere bei niedrigen Konzentrationen gegenüber den im Rahmen dieses Versuchs aufgenommen Messwerten auf. Für die Konzentrationen $c = 0.25\%, 0.5\%$ ist ein Vergleich der Regressionsparameter aus Plot 5 und Plot 4 nicht sinnvoll. Für die einprozentige Lösung fällt auf, dass die beiden Fitparameter α nicht miteinander verträglich sind. Im Betrage ist das α aus Plot 5 größer. Wie schon in Abschnitt erwähnt, ist es wahrscheinlich, dass sich bei dieser Probe das Guaran nicht vollständig im Wasser lösen konnte, da sich beim Anmischen der Lösung kleine Klumpen gebildet haben. Dies würde bedeuten, dass faktisch eine Konzentration $< 1\%$ untersucht wurde, was impliziert, dass der Fitparameter α, der für höhere Konzentrationen klarerweise im Betrage größer wird, im Betrage zu klein gemessen wurde. Für die Konzentration 1.45% in Plot 5 gibt es kein exaktes Gegenstück in Plot 4 . Es kann lediglich festgehalten werden, dass für die höhere Konzentration eine größeres $|\alpha|$ bestimmt wurde. Da die Konzentration von 1.4%, wie später anhand von Plot 7 noch erläutert wird, in einem Konzentrationsbereich liegt, in dem kleine Änderungen der Konzentration große Änderungen in der Viskosität nach sich ziehen, erscheint es auch plausibel, dass die beiden α nicht nahezu identisch sind, obwohl die Konzentrationen nahezu gleich sind.

Zuletzt kann der Scherstress gegen die Scherrate aufgetragen werden. Plot 6 enthält aber im Vergleich zu Plot 4 keinerlei neue Information, da die vom Rheometer ausgegebenen Messwerte für σ und η beide aus dem gemessenen Drehmoment berechnet wurden und zwischen ihnen die Abhängigkeit $\sigma = \dot{\gamma} \cdot \eta$ gilt.



	$c = 0.0\%$	$c = 0.25\%$	$c = 0.5\%$	$c = 1.0\%$	$c = 1.4\%$	$c = 2.0\%$	$c = 2.3\%$
α	0.257 ± 0.11	0.913 ± 0.25	1.197 ± 0.28	0.493 ± 0.021	0.3206 ± 0.014	0.2117 ± 0.014	0.1996 ± 0.017
β	-4.826 ± 0.29	-5.536 ± 0.67	-5.599 ± 0.76	0.7265 ± 0.059	2.4383 ± 0.039	3.8115 ± 0.037	4.2465 ± 0.045

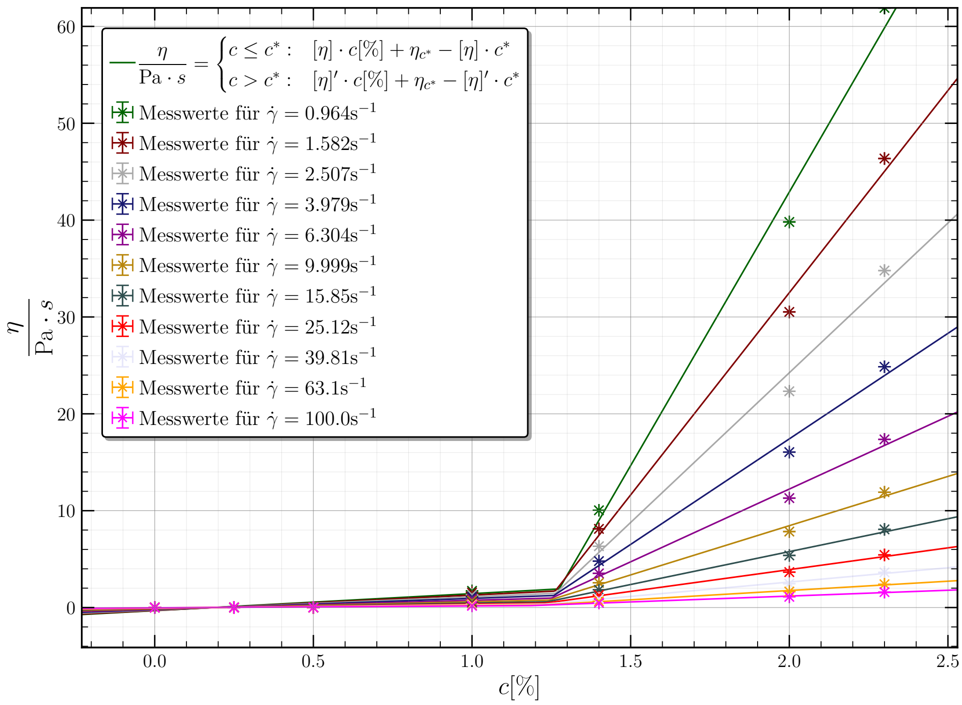
Plot 6 Doppelt logarithmische Auftragung von σ gegen $\dot{\gamma}$ bei Guaran

Entsprechend dieser Abhängigkeit gilt für die Parameter α_σ und α_η aus Plot 6 bzw. Plot 4 numerisch exakt $\alpha_\sigma - \alpha_\eta = 1$. Da $0 < \alpha_\sigma < 1$ kann erneut bestätigt werden, dass scherverdünnendes Verhalten vorliegt.

Schon rein visuell fällt in Plot 4 auf, dass für die Viskositäten zwischen den Konzentrationen $c = 1.0\%$ und $c = 1.4\%$ gemessen an der kleinen Konzentrationsdifferenz ein recht großer Sprung in der Viskosität zu beobachten ist. Und tatsächlich modelliert man die Abhängigkeit der Viskosität von der Konzentration bei Polymerlösungen häufig wie folgt:

$$\eta(c) = \begin{cases} c \leq c^* : & [\eta] \cdot (c - c^*) + \eta_{c^*} \\ c > c^* : & [\eta]' \cdot (c - c^*) + \eta_{c^*} \end{cases} \tag{11}$$

Für eine feste Scherrate kann somit durch Variation der Parameter $([\eta], [\eta]', c^*, \eta_{c^*})$ ein Fit der Daten vorgenommen werden. Es ergibt sich der folgende Plot:



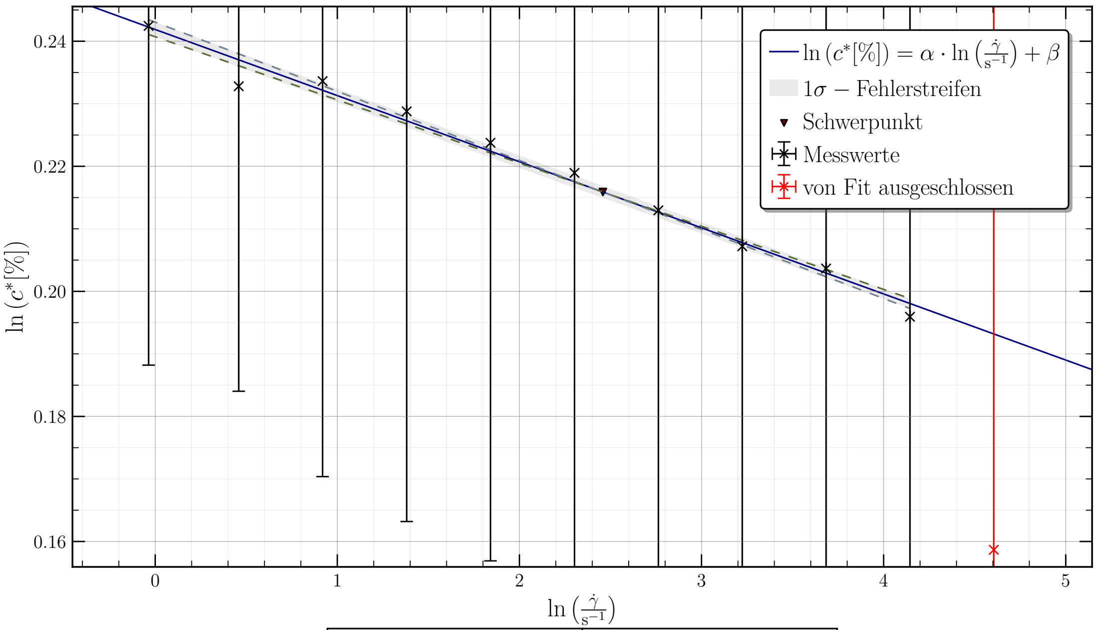
	$\dot{\gamma} = 0.964\mathrm{s}^{-1}$	$\dot{\gamma} = 1.582\mathrm{s}^{-1}$	$\dot{\gamma} = 2.507\mathrm{s}^{-1}$	$\dot{\gamma} = 3.979\mathrm{s}^{-1}$	$\dot{\gamma} = 6.304\mathrm{s}^{-1}$	$\dot{\gamma} = 9.999\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$c^*[\%]$	1.274 ± 0.07	1.262 ± 0.06	1.263 ± 0.08	1.257 ± 0.08	1.251 ± 0.08	1.245 ± 0.08	1.237 ± 0.08	1.23 ± 0.09	1.226 ± 0.09	1.216 ± 0.09	1.17 ± 0.1
$\frac{\eta_{c^*}}{\mathrm{Pa \cdot s}}$	1.91 ± 3	1.7 ± 2	1.46 ± 2	1.22 ± 1	0.98 ± 0.9	0.77 ± 0.6	0.59 ± 0.4	0.45 ± 0.3	0.34 ± 0.2	0.25 ± 0.1	0.182 ± 0.09
$\frac{[\eta]}{\mathrm{Pa \cdot s}}$	1.75 ± 3	1.59 ± 2	1.36 ± 2	1.15 ± 1	0.92 ± 1	0.73 ± 0.6	0.57 ± 0.4	0.43 ± 0.3	0.33 ± 0.2	0.24 ± 0.1	0.18 ± 0.09
$\frac{[\eta]'}{\mathrm{Pa \cdot s}}$	56.49 ± 3	41.75 ± 2	30.91 ± 2	21.79 ± 2	15.02 ± 1	10.17 ± 0.7	6.79 ± 0.5	4.5 ± 0.3	2.96 ± 0.2	1.92 ± 0.1	1.21 ± 0.1

Plot 7 Auftragung von η gegen c bei Guaran

In der Appendix A.2 sind für jede Scherrate dieselben Daten nochmals in einzelnen Figures aufgetragen, um die Übersichtlichkeit zu verbessern. Qualitativ kann festgehalten werden, dass das verwendete Modell die Messdaten gut zu beschreiben vermag, also tatsächlich eine Konzentration c^* existiert, ab der der Anstieg der Viskosität mit der Konzentration schneller wird.

Die Güte der Fits mag überraschend sein, da bei der Betrachtung der Viskosität als Funktion der Scherrate zumindest für zwei Konzentrationen äußerst unzufriedenstellende Messwerte aufgenommen wurden. Da aber für jeden Fit $\dot{\gamma}$ fest ist wird, haben wir bei

jedem der 11 Fits in Plot 7 mindestens vier Messwerte mit guter Ausagekraft, sodass diese den Einfluss der beobachteten Abweichung der zwei bis drei anderen Messwerte abschwächen und zu einer relativ hohen Aussagekraft der Fitparameter führen.
Ein Blick auf die Fitparameter zeigt, dass c^* monoton mit der Scherrate $\dot{\gamma}$ abzunehmen scheint. Tatsächlich zeigt sich in einer doppelt logarithmischen Darstelllung der kritischen Konzentration gegen die Scherrate ein klarer linearer Trend ($\rightarrow$ Plot 8). Beim Wert zu $\dot{\gamma} = 100\ \mathrm{s}^{-1}$ handelt es sich um einen statistischen Ausreißer, der deshalb nicht bei der Regression berücksichtigt wurde. Ein Berücksichtigen dieses Messwertes hätte zu einer Verschlechterung in der Beschreibung des ansonsten eindeutig linearen Zusammenhangs geführt.



	Wert	Unsicherheit
α	-0.010572	0.0004732
β	0.241874	0.001163

Plot 8 Doppelt logarithmische Auftragung von c^* gegen $\dot{\gamma}$ bei Guaran

In der Tat erscheint es plausibel, dass ein Potenzgesetz zwischen der kritischen Konzentration und der Scherrate vorliegt: Die Viskosität nimmt gemäß eines Potenzgesetzes mit der Scherrate ab. Das heißt auch, dass die Kurven in Plot 7 für festes c gemäß

$$\frac{\dot{\gamma}_1}{\dot{\gamma}_2} = \frac{\eta_1^{\alpha(c)}}{\eta_2^{\alpha(c)}} \tag{12}$$

ineinander überführt werden können. Stellt man sich nun eine Kurve (η, c) für festes $\dot{\gamma} = \dot{\gamma}_2$ vor und möchte diese in eine Kurve mit $\dot{\gamma} = \dot{\gamma}_1 > \dot{\gamma}_2$ überführen, so gilt $\eta_1 =$

$\left(\frac{\dot{\gamma}_1}{\dot{\gamma}_2}\right)^{1/\alpha(c)} \cdot \eta_2$. Nun kann man Plot 4 entnehmen, dass $|\alpha|$ für größere Konzentrationen zunimmt. Das bedeutet auch, dass die η-Werte für kleinere c (relativ gesehen) weniger gestaucht werden als die Werte mit größere c. Damit flacht die Kurve $\eta_(c)\big|_{\dot{\gamma}_1}$ im Vergleich zur Kurve $\eta_(c)\big|_{\dot{\gamma}_2}$ ab, und insbesondere wird der Punkt der kritischen Konzentration nach links verschoben (zu kleineren c). Für die konkrete Form des Potenzgesetzes, wie c^* von $\dot{\gamma}$ abhängt, kommt es nun auf die konkrete Form der Abhängigkeit $\alpha(c)$ an. Klar ist aber, dass zwischen c^* und $\dot{\gamma}$ ein Potenzzusammenhang gelten muss.

Da also, wie gerade dargelegt, die kritische Konzentration nicht nur eine Stoffeigenschaft ist, sondern auch von der Scherrate abhängt, ist es nicht sinnvoll, abschließend *eine* kritische Konzentration oder Überlappungskonzentration für Guaran-Wasser-Lösungen festzuhalten. Stattdessen bemerken wir, dass der folgende funktionale Zusammenhang empirisch gefunden wurde:

$$c^*[\%] = e^{0.242\pm0.002} \cdot \dot{\gamma}^{-0.0106\pm0.0005} \tag{13}$$

Wir haben keinen Wert in der Literatur gefunden, mit dem dieses Ergebnis sinnvoll verglichen werden könnte. Dies könnte darin begründet liegen, dass die Abhängigkeit c^* von $\dot{\gamma}$ ein Artefakt einer unzulänglichen Versuchsdurchführung sein könnte und die kritische Konzentration c^* möglicherweise eigentlich eine Konstante ist, die nur vom gelösten Stoff abhängt. Gestützt wird dies dadurch, dass der Exponent im bestimmten Potenzgesetz nahe 0 ist.

4.2.2. Fehlerabschätzung zum Anmischen der Lösungen

Analog zu den Überlegungen in Abschnitt 4.1.2 können die Unsicherheiten für die Konzentrationen bei den Guaran-Lösungen abgeschätzt werden. Man erhält:

c[%]-Soll	$V_{H_2O}[\mu l]$	m[g]	$c = \frac{m}{\rho \cdot V + m}$[%]	Δc[%]	$\frac{\Delta c}{c}$[%]
0.25	1189	0.00298	0.2507	0.01680	6.6984
0.5	1465	0.00736	0.5013	0.01359	2.7113
1	1096	0.01107	1.0029	0.01809	1.8038
1.4	1158	0.01644	1.4039	0.01714	1.2205
2	963	0.01964	2.0045	0.02063	1.0291
2.3	1078	0.02537	2.3060	0.01856	0.8047

Man erkennt, dass die relativen Fehler in der Konzentration sehr klein werden für größere Konzentrationen. Die Konzentrationen hatten also im Rahmen der geringen Messunsicherheit die Soll-Konzentration der jeweiligen Stoffe, sodass eine zu große oder zu kleine untersuchte Konzentration wegen ungenau abgemessener Massen oder Volumina zumindest bei Konzentrationen $\geq 1\%$ keine relevante Fehlerquelle ist. Für die kleinere Konzentrationen die Stoffkonzentration muss die Unsicherheit in der Stoffkonzentration jedoch prinzipiell als statistische Fehler berücksichtigt werden.

4.2.3. Fehlerbetrachtung Scherratenmessungen

Wie in im vorigen Abschnitt 4.2.1 bereits angemerkt, muss das beobachtete Verhalten bei $c = 0.25\%, 0.5\%$ auf systematische Messfehler zurückzuführen sein. Es sind folgende Fehlereinflüsse denkbar:

- Under- oder Overfilling: In besonderem Maße wurde bei dem Loaden des Samples bei den Guar-Messungen darauf geachtet, dass eine ausreichende Lösungsmenge aufgebracht wird, da stets ein Teil der mittels einer Pipette abgemessenen Menge in der Pipette zurückblieb. Hellström (2015) merkt an: *„Small changes in the radius of a rheometer sample can cause significant errors in the measured apparent viscosity* “ ([8]). Er empfiehlt ein Verfahren, bei dem mittels Bildgebung der reale Sample-Radius bestimmt wird, um auf diesen Fehler hin zu korrigieren. Dies ist auch bei dem hier verwendeten Setup leicht umzusetzen und verspricht erhebliche Verbesserungen in der Messgenauigkeit.

- Nicht-Konstanz der Sample-Temperatur während der Messung (vgl. dazu auch Abschnitt 4.1.1): Kleine Variationen in der Temperatur führen zu vernachlässigbaren Messfehlern. Bei den Saccharose-Messungen wurde jedoch schon die Beobachtung gemacht, dass - entgegen der Erwartung- die Saccharose-Lösungen und sogar die reine Wasserlösung scherverdünnendes Verhalten zeigen. Da die Temperatur nicht direkt am Sample gemessen wird (sonder in der Platte darunter), und da die Anfangstemperatur des Samples etwas 12 K unter der Soll-Temperatur während des Versuchs lag, bleibt die Frage offen, ob das Sample bei der ersten Scherrate für die bereits die Soll-Temperatur erreicht hatte. Würde sich die Temperatur des Samples im Verlauf der weiteren Messungen weiter erhöhen, würde ein verstärkt scherverdünnendes Verhalten beobachtet werden, da höhere Temperaturen zu einer Abnahme der Viskosität führen.

- (kleine) Luftblaseneinschlüsse im Sample, der beim Anmischen der Lösungen entstanden sein könnte: Stadler spekuliert, dass die wegen ihrer Oberflächenspannung sehr elastischen Luftblasen bei Elastizitätsmessungen deformiert werden und zu einer erhöhten Elastiziätsmessung führen ([6]). Analoge Überlegugnen könnten nahelegen, dass Lufblaseneinschlüsse zu einer geringeren Viskosität führen, da die Luftblasen lokal des Zusammenhalt der Polymermoleküle schwächen würden. Denkbar ist auch, dass sich bei einer gewissen Scherrate zwei eingeschlossene Luftblasen vereinigen, da sie sich durch die Deformation näher kommen. Ein solches Vereinigen könnte zu einem unstetigen Verhalten bei der Viskositätsmessung führen. Die Bewegung von Luftblasen in Flüssigkeiten ist hydrodynamisch jedoch sehr komplex, sodass der konkrete Einfluss schwer abzuschätzen ist. Klar ist jedoch, dass Luftblaseneinschlüsse im Sample, wie sie beim Schütteln des Samples oder dem Pipettieren entstanden sein könnten, als wahrscheinlich eingestuft werden müssen.

- nicht vollständig gelöstes Guaran-Pulver, das ausklumpt: Solche Klumpen würden eine Inhomogenität in der Lösung bedeuten, die lokal die Viskosität (stark) erhöht. Insgesamt wäre die gelöste Pulvermenge durch solche Klumpen verringert, da das

Pulver innerhalb der Klumpen natürlich nicht gelöst werden kann. Das Auftreten solcher Klumpen steht außer Frage, jedoch wurden sie stets so weit ausgemerzt, bis keine mehr mit bloßem Auge sichtbar waren. Bei der 1%igen Lösung sind solche Einflüsse wegen Schwierigkeiten beim Anmischen wahrscheinlich. Dort ist jedoch im Gegensatz zu den Lösungen mit $c = 0.25\%, 0.5\%$ ein klar linearer Zusammenhang erkennbar, weshalb Klumpenbildung vermutlich nicht alleine das Verhalten bei den besagten zwei Konzentrationen erklären kann.

- Erschütterungen im Raum während der Versuchsdurchführung: Einflüsse dadurch sind wegen der hohen Genauigkeit bei der Drehmomentbestimmung prinzipiell möglich, beispielsweise wenn die Tür rapide zugezogen werden würde. Äußern würde es sich durch einzelne Messausreißer.
  Für die Messungen mit $c = 0.25\%, 0.5\%$ kann ein solcher Einfluss nicht als Erklärung herangezogen werden, da die beiden Messungen in dem Sinn korrelliert erscheinen, dass sie das gleiche qualitative Verhalten (aus zwei nach unten gerichteten Hyperbelästen zu bestehen) zeigen.

- Falschausrichtung der Geometry (vgl. dazu [6]): Wurde die Geometry in der Vergangenheit einmal fallen gelassen oder hat sie sich anderweitig deformiert, ist es denkbar, dass die beiden Platten nicht mehr parallel sind. Dies würde den Plattenabstand (Gap) verändern, was die ausgegebenen Scherraten wegen $\dot{\gamma} \propto d^{-1}$ systematisch verfälschen würde. Da das verwendete Rheometer jedoch häufig in Benutzung ist, ist es wahrscheinlich, dass solche Veränderungen sehr frühzeitig bemerkt werden würden und deshalb der Einfluss von dadurch induzierten Fehlern als unwahrscheinlich einzuschätzen.

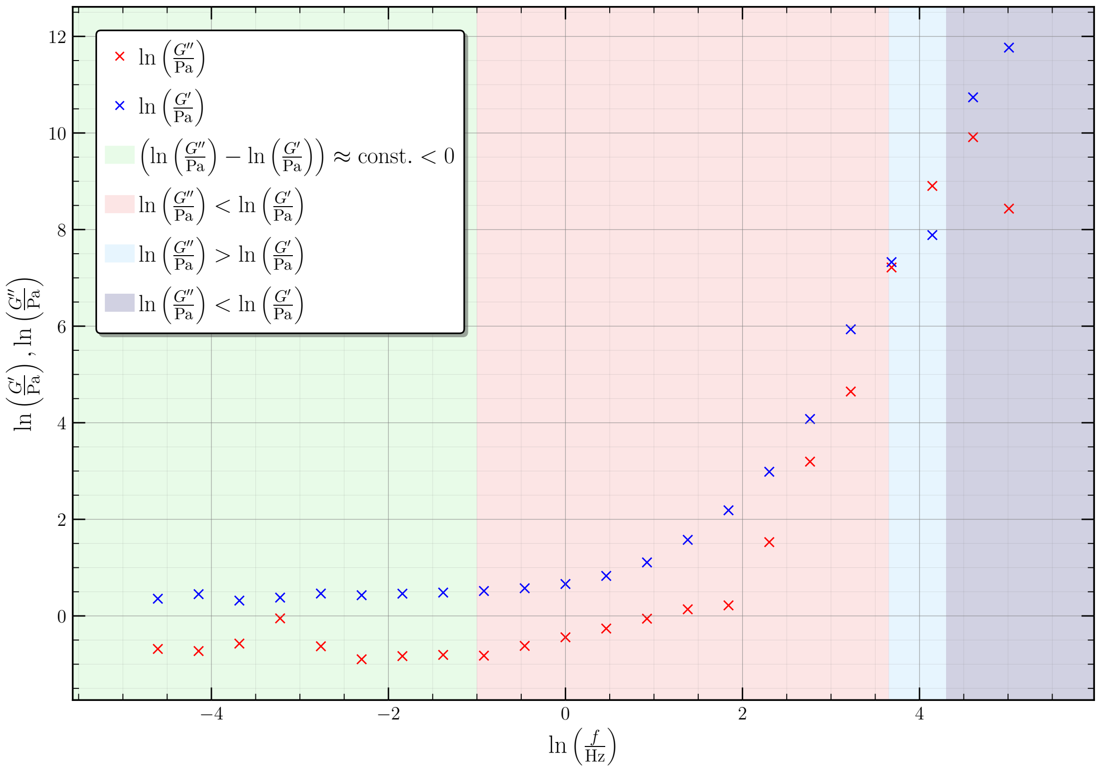
Außerdem gilt weiterhin die Bemerkung aus Abschnitt 2.4, dass prinzipiell der relative Fehler bei der Viskositätsbestimmung für kleiner Viskositäten und somit kleinere Stoffkonzentrationen größer ist. Dies Zusammen mit der Nicht-Konstanz der Temperatur und der Gefahr von Under- oder Overfilling sind als die wahrscheinlichsten Fehlereinflüsse zu nennen. Zuletzt sei folgende Bemerkung von Stadler ([6]) erwähnt, wonach ein erheblicher Anteil der beobachteten Anomalitäten auf falsche Anwendung seitens des Benutzers zurückzuführen sind:

„many inexperienced users obtain data, which is significantly error afflicted in nontrivial ways“

Denkbar hier sind das unzureichende Reinigen der Platten am Rheometer vor dem Loaden des Samples oder sonstige Flüchtigkeitsfehler.

4.2.4. Frequenzversuch

Trägt man das Speicher- bzw. Verlustmodul doppellogarithmisch gegen die Oszillationsfrequenz auf, erhält man den folgenden Plot, in dem Frequenzbereiche, die qualitativ unterschiedliches Verhalten zeigen, durch farbliche Untermalung hervorgehoben sind (die Grenzen der Bereiche sind nicht als scharf zu sehen, da sie nur „per Augenmaß“ eingezeichnet wurden):



Plot 9 Doppelt logarithmische Auftragung von G', G'' gegen die Frequenz f bei Guaran

Es zeigt sich, dass für kleine Frequenzen sowohl das Speicher- als auch das Verlustmodul in sehr guter Näherung konstant sind, wobei das Speichermodul größer ist als das Verlustmodul (grün hinterlegt). In Abbildung 5 kann dies der Rubbery Plateau Region zugeordnet werden, in der sich der Stoff gelartig verhält.
Im sich anschließenden rot hinterlegten Bereich nehmen die viskosen Anteile weiter zu, bis sich das Speicher- und das Verlustmodul bei $f^* \approx 40$ Hz kreuzen. Für einen kurzen Frequenzbereich (hellblau hinterlegt) dominiert das Verlustmodul gegenüber dem Speichermodul. Dieser Bereich kann der Transition Region in Abbildung 5 zugeordnet werden. In diesem Bereich wäre eine höhere Messwertdichte wünschenswert, um eine definitivere Aussage darüber treffen zu können, in welchem Bereich $G'' > G'$ ist. Anschließend nimmt das Speichermodul weiter zu und das Verlustmodul ab, sodass $G'' > G'$ (dunkelblau hinterlegt). Gemäß Abbildung 5 ist für noch höhere Frequenzen ein Abflachen von G' und Abnehmen von G'' zu erwarten. Es wäre schön, noch weitere Messwerte zu sehen, um das Verhalten, das vermutlich der Glossy Region zugeordnet werden kann, weiter zu beleuchten.
Ferner konnte für keinen Frequenzbereich ein Verhalten wie in der Terminal Region beobachtet werden. Es ist möglich, dass für kleinere Frequenzen ($\lesssim 0.01$ Hz) als im betrachteten Messbereich ein solches auftreten würde.
Insgesamt konnten also drei der vier Regionen, die im viskoelastischen Spektrum einer

Polymerlösung zu erwarten sind, bei der Analyse der 0.5%igen Guaran-Wasser-Lösung beobachtet werden.

4.2.5. Fehlerbetrachtung Frequenzversuch

Da die Beobachtungen, die zum Frequenzversuch gemacht wurden, nur qualitativer Natur sind, ist der Einfluss von systematischen Fehlern als äußerst gering einzustufen. Eine von der Soll-Temperatur verschiedene Temperatur würde die Breite und Höhe der einzelnen viskoelastischen Bereiche verändern, aber nichts an deren qualitativen Natur ändern. Selbst eine sich zeitlich (monoton) ändernde Temperatur des Samples, wie in Abschnitt 4.1.1 diskutiert, würde den qualitativen Verlauf kaum ändern, sondern nur zu einer Stauchung entlang der Abszisse und auch Ordinate führen und damit zwar u.U. das Bestimmen von f^* systematisch verfälschen, aber an der prinzipiellen Abfolge der viskoelastischen Bereiche nichts ändern. Der Einfluss durch Temperaturvariation ist jedoch (im Gegensatz zu den Messungen mit Saccharose) ohnehin als unerheblich einzustufen, da die 0.5%ige Lösung wegen der zuvor mit ihr vorgenommenen Messungen schon lange vor dem Beginn der eigentlichen Oszillationsmessung auf das Rheometer gegeben wurde, also auf jeden Fall genug Zeit hatte, die Temperatur Soll-Temperatur von $25°C$ anzunehmen.

Auch der Einfluss eines Underfills wäre nur gering, da dadurch die elastischen und viskosen Eigenschaften gleichermaßen als zu gering gemessen worden wären, der Graph dadurch also nur entlang der Ordinate gestaucht werden würde.

Der größte Fehler bei der Bestimmung der einzelnen viskoelastischen Bereiche ist die recht grobe Auflösung in der Frequenz, da dadurch beispielsweise die Schnittpunkte vom Verlust- und vom Speichermodul nur äußerst grob abgeschätzt werden konnten. Es wäre sinnvoll, die Messpunkte pro Dekade bei einer Wiederholung des Versuchs zu Kosten einer etwas längeren Versuchsdurchführung auf einen Wert > 5 zu setzen. Auch ist es empfehlenswert, den Messbereich selbst zu vergrößern. Bei einigen hundert Hz maximaler Messfrequenz mehr könnte die Glossy Region deutlich detaillierter beleuchtet werden. Änderte man zudem die minimale Messfrequenz auf einen kleineren Wert, ist es denkbar, dass die Terminal Region noch zu beobachten wäre.

5. Zusammenfassung

Das beobachtete Verhalten, wonach die Saccharoselösungen scheinbar scherverdünnendes Verhalten zeigen, kann über eine systematische Verfälschung der Messwerte durch ein Ansteigen der Temperatur während des Messprozesses plausibel erklärt werden. Leider gelang es deshalb nicht, abschließend zuverlässige Werte für die Viskositäten von den Saccharoselösungen oder einen Wert für die intrinsische Viskosität von Saccharose festzuhalten. Ein ausschlaggebender Fehlereinfluss hierfür war, selbst nachdem die Temperaturfehler durch ausschließliches Berücksichtigen der Messwerte für hohe Scherraten, mit großer Sicherheit ein Underfilling. Wegen des ausdrücklichen Hinweises seitens des Betreuers, bei den Guaran-Lösungen lieber etwas zu viel als zu wenig Probe ins Rheometer zu geben, wurde war Underfilling wahrscheinlich bei den Guaran-Messungen ein geringerer Einflussfaktor als bei den Saccharose-Messungen. Mit der im Versuchsverlauf gewonnenen Erfahrung der Versuchsdurchführenden wäre ein Wiederholen der Saccharose-Messungen sinnvoll. Die dann gewonnenen Daten würden mit Sicherheit bestätigen, dass sich Saccharoselösungen in der Tat als Newtonsche Flüssigkeiten verhalten, und ferner zulassen, die (intrinsische) Viskosität von Saccharose-Wasser-Lösungen genau zu bestimmen.
Es gelingt uns nicht, eine zufriedenstellende Erklärung für die Messwerte bei den Konzentrationen $c = 0.25\%, 0.5\%$ zu finden. Ein Wiederholen dieser Messungen wird ausdrücklich empfohlen, um zu prüfen, ob das beobachtete Verhalten reproduzierbar ist. Zu erwarten ist jedoch, dass sich bei einer Wiederholung „schönere“ Werte ergeben würden, die erstens stützen, dass auch bei diesen Konzentrationen zwischen der Viskosität und der Scherrate ein Potenzgesetzzusammenhang gilt, und zweitens, das Guaran-Wasser-Lösungen für beliebige Konzentrationen scherverdünnend sind.
Weiterhin empfehlen wir eine Versuchsdurchführung in einem wärmeren Raum. Wahrscheinlich würde dann kein scherverdünnendes Verhalten bei den Saccharose-Lösungen beobachtet werden.

Literatur

[1] Autor unbekannt, “Praktikumsversuch Rheologie Bachelor.” [Online unter https://www.softmatter.physik.uni-muenchen.de/teaching/fortgeschrittenenpraktikum/r3_rheologie/fpraktikumrheologiebdeutsch.pdf; Stand 03. Dezember 2022].

[2] Wikipedia, “Potenzgesetz (Flüssigkeit).” https://de.wikipedia.org/w/index.php?title=Potenzgesetz_(Fl%C3%BCssigkeit)&oldid=192899581, 2019. [Online; Stand 3. Dezember 2022].

[3] Malvern, “Kinexus series user manual,” Jan 2014. [Online unter https://www.equipx.net/uploads/Malvern%20Instruments/MalvernKinexus-usermanual.pdf, Stand 08.12.2022].

[4] Malvern, “Kinexus Brochure,” 2017. [Online unter https://www.malvernpanalytical.com/de/assets/MRK1089-06-DE_Kinexus_Brochure_LRA4_tcm57-17219.pdf, Stand 08.12.2022].

[5] P. Först, “In-situ Untersuchungen der Viskosität fluider, komprimierter Lebensmittel-Modellsysteme,” 2001.

[6] F. J. Stadler, “What are typical sources of error in rotational rheometry of polymer melts?,” *Korea-Australia Rheology Journal*, vol. 26, pp. 277–291, Aug 2014.

[7] V. Telis, J. Telis-Romero, H. Mazzotti, and A. Gabas, “Viscosity of Aqueous Carbohydrate Solutions at Different Temperatures and Concentrations,” *International Journal of Food Properties*, vol. 10, no. 1, pp. 185–195, 2007.

[8] L. H. O. Hellström, M. A. Samaha, K. M. Wang, A. J. Smits, and M. Hultmark, “Errors in parallel-plate and cone-plate rheometer measurements due to sample underfill,” *Measurement Science and Technology*, vol. 26, p. 015301, nov 2014.

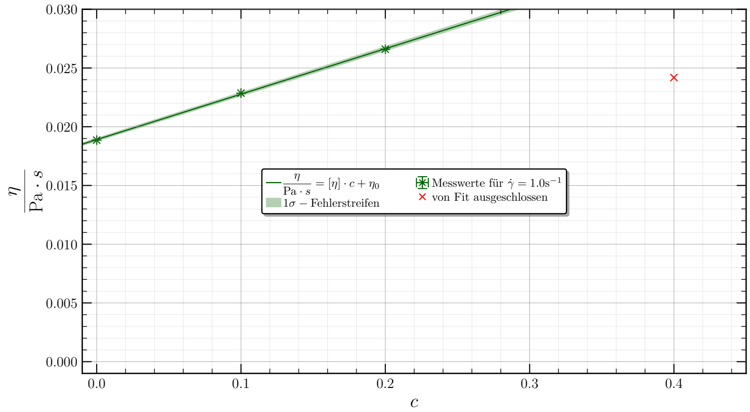
[9] E. W. Lemmon, I. H. Bell, M. L. Huber, M. O. McLinden, P. Linstrom, and W. Mallard, *Thermophysical Properties of Fluid Systems.* National Institute of Standards and Technology. [Online unter https://doi.org/10.18434/T4D303, Zugriff am 11.12.2022].

[10] Wikipedia contributors, “Guar gum.” https://en.wikipedia.org/w/index.php?title=Guar_gum&oldid=1124053245, 2022. [Online; Zugriff am 08. Dezember 2022].

[11] J. Kramer, J. T. Uhl, and R. K. Prud’ Homme, “Measurement of the viscosity of guar gum solutions to 50,000 s-1 using a parallel plate rheometer,” *Polymer Engineering & Science*, vol. 27, no. 8, pp. 598–602, 1987.

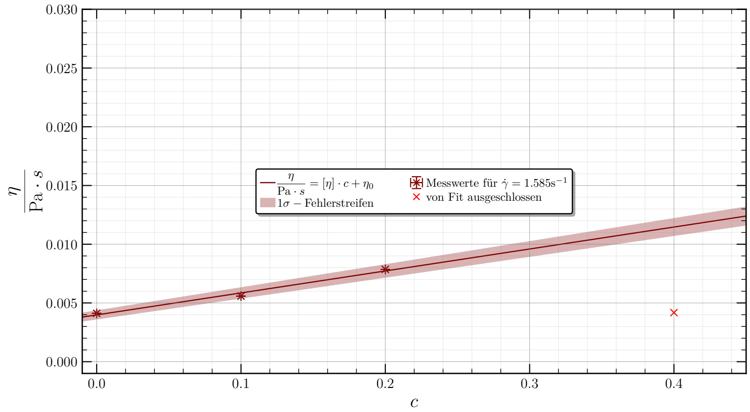
A. Ergänzende Plots

A.1. Bestimmung der Konzentrationsabhängigkeit zu Saccharose



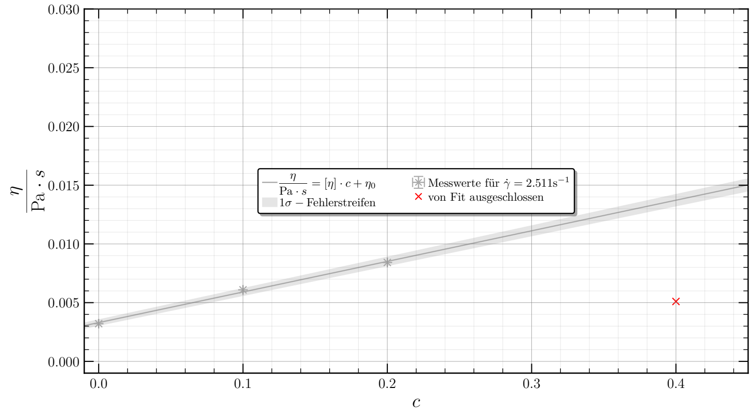
	$\dot{\gamma} = 1.000\mathrm{s}^{-1}$	$\dot{\gamma} = 1.585\mathrm{s}^{-1}$	$\dot{\gamma} = 2.511\mathrm{s}^{-1}$	$\dot{\gamma} = 3.981\mathrm{s}^{-1}$	$\dot{\gamma} = 6.310\mathrm{s}^{-1}$	$\dot{\gamma} = 10.000\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$\frac{[\eta]}{\mathrm{mPa}}$	387.0 ± 7.5	187.0 ± 24	260.5 ± 16	120.0 ± 23	37.0 ± 33	32.55 ± 34	0.95 ± 15	-27.8 ± 0.52	-29.5 ± 3.6	-28.96 ± 1.2	12.05 ± 14
$\frac{\eta_0}{\mathrm{mPa}}$	189.033 ± 0.97	39.83 ± 3.1	33.02 ± 2	23.43 ± 3	19.8 ± 4.2	9.34 ± 4.4	9.84 ± 2	11.53 ± 0.067	8.417 ± 0.46	6.162 ± 0.16	0.76 ± 1.8

Plot 10 Auftragung von η gegen c bei Saccharose für Scherrate 1



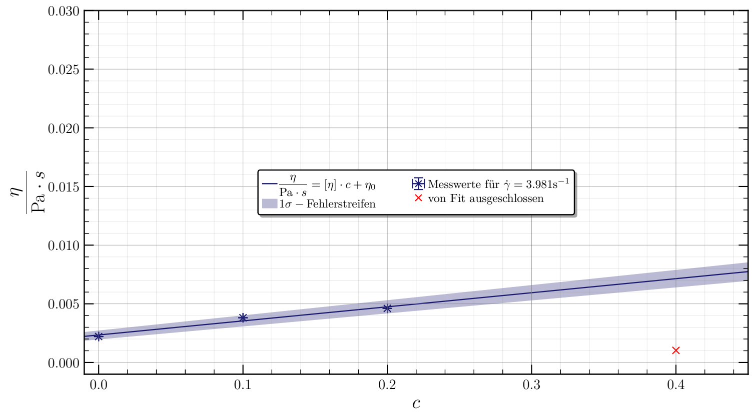
	$\dot{\gamma} = 1.000\mathrm{s}^{-1}$	$\dot{\gamma} = 1.585\mathrm{s}^{-1}$	$\dot{\gamma} = 2.511\mathrm{s}^{-1}$	$\dot{\gamma} = 3.981\mathrm{s}^{-1}$	$\dot{\gamma} = 6.310\mathrm{s}^{-1}$	$\dot{\gamma} = 10.000\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$\frac{[\eta]}{\mathrm{mPa}}$	387.0 ± 7.5	187.0 ± 24	260.5 ± 16	120.0 ± 23	37.0 ± 33	32.55 ± 34	0.95 ± 15	-27.8 ± 0.52	-29.5 ± 3.6	-28.96 ± 1.2	12.05 ± 14
$\frac{\eta_0}{\mathrm{mPa}}$	189.033 ± 0.97	39.83 ± 3.1	33.02 ± 2	23.43 ± 3	19.8 ± 4.2	9.34 ± 4.4	9.84 ± 2	11.53 ± 0.067	8.417 ± 0.46	6.162 ± 0.16	0.76 ± 1.8

Plot 11 Auftragung von η gegen c bei Saccharose für Scherrate 2



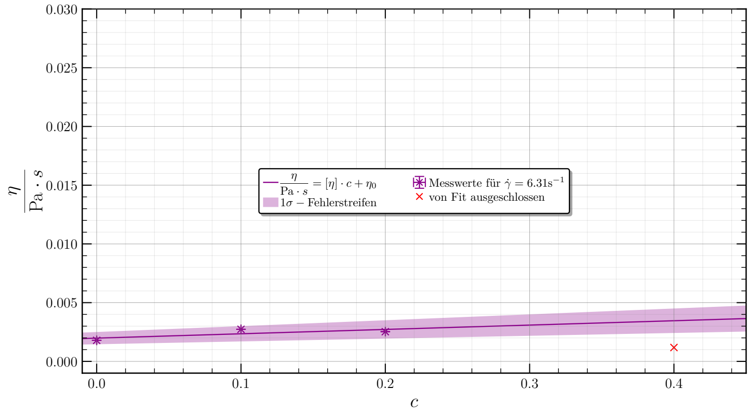
	$\dot{\gamma} = 1.000\mathrm{s}^{-1}$	$\dot{\gamma} = 1.585\mathrm{s}^{-1}$	$\dot{\gamma} = 2.511\mathrm{s}^{-1}$	$\dot{\gamma} = 3.981\mathrm{s}^{-1}$	$\dot{\gamma} = 6.310\mathrm{s}^{-1}$	$\dot{\gamma} = 10.000\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$\frac{[\eta]}{\mathrm{mPa}}$	387.0 ± 7.5	187.0 ± 24	260.5 ± 16	120.0 ± 23	37.0 ± 33	32.55 ± 34	0.95 ± 15	-27.8 ± 0.52	-29.5 ± 3.6	-28.96 ± 1.2	12.05 ± 14
$\frac{\eta_0}{\mathrm{mPa}}$	189.033 ± 0.97	39.83 ± 3.1	33.02 ± 2	23.43 ± 3	19.8 ± 4.2	9.34 ± 4.4	9.84 ± 2	11.53 ± 0.067	8.417 ± 0.46	6.162 ± 0.16	0.76 ± 1.8

Plot 12 Auftragung von η gegen c bei Saccharose für Scherrate 3



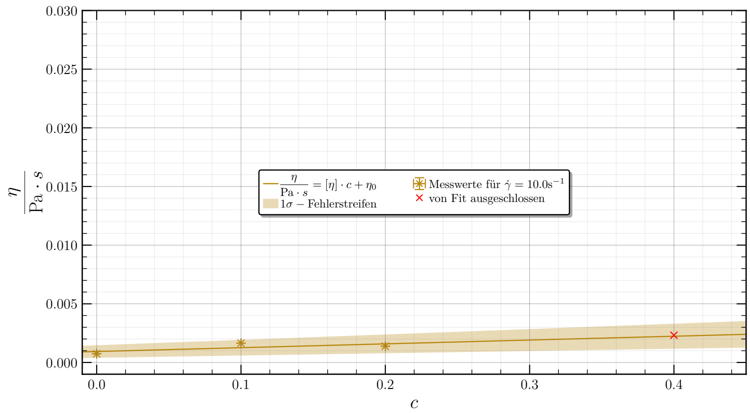
	$\dot{\gamma} = 1.000\mathrm{s}^{-1}$	$\dot{\gamma} = 1.585\mathrm{s}^{-1}$	$\dot{\gamma} = 2.511\mathrm{s}^{-1}$	$\dot{\gamma} = 3.981\mathrm{s}^{-1}$	$\dot{\gamma} = 6.310\mathrm{s}^{-1}$	$\dot{\gamma} = 10.000\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$\frac{[\eta]}{\mathrm{mPa}}$	387.0 ± 7.5	187.0 ± 24	260.5 ± 16	120.0 ± 23	37.0 ± 33	32.55 ± 34	0.95 ± 15	-27.8 ± 0.52	-29.5 ± 3.6	-28.96 ± 1.2	12.05 ± 14
$\frac{\eta_0}{\mathrm{mPa}}$	189.033 ± 0.97	39.83 ± 3.1	33.02 ± 2	23.43 ± 3	19.8 ± 4.2	9.34 ± 4.4	9.84 ± 2	11.53 ± 0.067	8.417 ± 0.46	6.162 ± 0.16	0.76 ± 1.8

Plot 13 Auftragung von η gegen c bei Saccharose für Scherrate 4



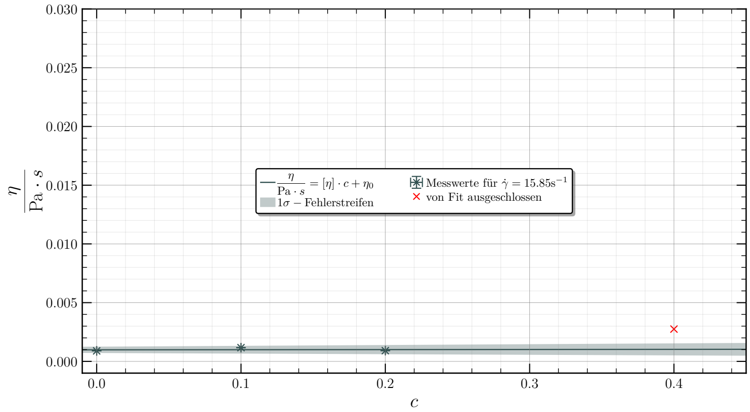
	$\dot{\gamma} = 1.000\mathrm{s}^{-1}$	$\dot{\gamma} = 1.585\mathrm{s}^{-1}$	$\dot{\gamma} = 2.511\mathrm{s}^{-1}$	$\dot{\gamma} = 3.981\mathrm{s}^{-1}$	$\dot{\gamma} = 6.310\mathrm{s}^{-1}$	$\dot{\gamma} = 10.000\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$\frac{[\eta]}{\mathrm{mPa}}$	387.0 ± 7.5	187.0 ± 24	260.5 ± 16	120.0 ± 23	37.0 ± 33	32.55 ± 34	0.95 ± 15	-27.8 ± 0.52	-29.5 ± 3.6	-28.96 ± 1.2	12.05 ± 14
$\frac{\eta_0}{\mathrm{mPa}}$	189.033 ± 0.97	39.83 ± 3.1	33.02 ± 2	23.43 ± 3	19.8 ± 4.2	9.34 ± 4.4	9.84 ± 2	11.53 ± 0.067	8.417 ± 0.46	6.162 ± 0.16	0.76 ± 1.8

Plot 14 Auftragung von η gegen c bei Saccharose für Scherrate 5



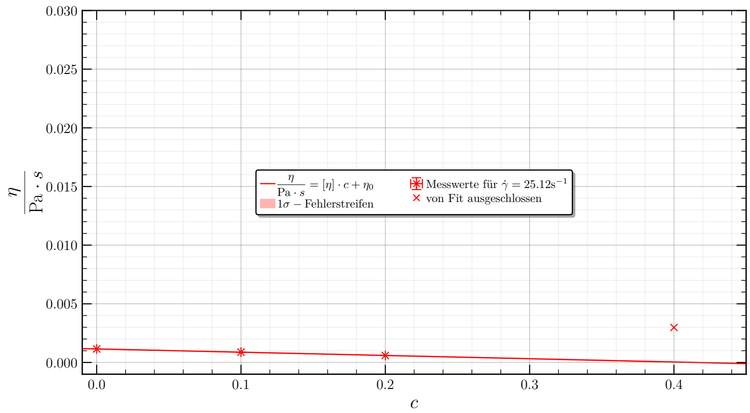
	$\dot{\gamma} = 1.000\mathrm{s}^{-1}$	$\dot{\gamma} = 1.585\mathrm{s}^{-1}$	$\dot{\gamma} = 2.511\mathrm{s}^{-1}$	$\dot{\gamma} = 3.981\mathrm{s}^{-1}$	$\dot{\gamma} = 6.310\mathrm{s}^{-1}$	$\dot{\gamma} = 10.000\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$\frac{[\eta]}{\mathrm{mPa}}$	387.0 ± 7.5	187.0 ± 24	260.5 ± 16	120.0 ± 23	37.0 ± 33	32.55 ± 34	0.95 ± 15	-27.8 ± 0.52	-29.5 ± 3.6	-28.96 ± 1.2	12.05 ± 14
$\frac{\eta_0}{\mathrm{mPa}}$	189.033 ± 0.97	39.83 ± 3.1	33.02 ± 2	23.43 ± 3	19.8 ± 4.2	9.34 ± 4.4	9.84 ± 2	11.53 ± 0.067	8.417 ± 0.46	6.162 ± 0.16	0.76 ± 1.8

Plot 15 Auftragung von η gegen c bei Saccharose für Scherrate 6



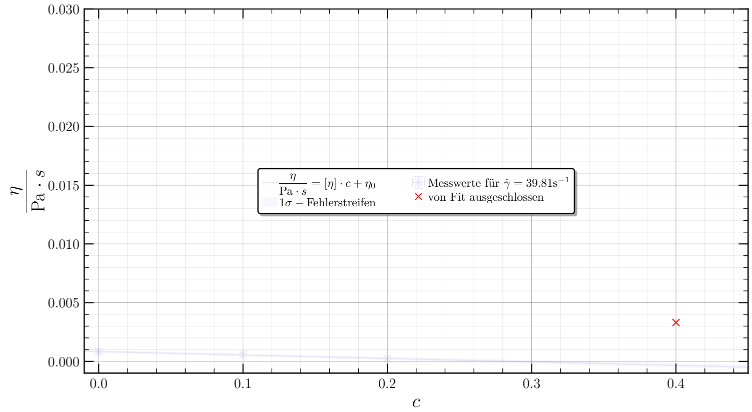
	$\dot{\gamma} = 1.000\mathrm{s}^{-1}$	$\dot{\gamma} = 1.585\mathrm{s}^{-1}$	$\dot{\gamma} = 2.511\mathrm{s}^{-1}$	$\dot{\gamma} = 3.981\mathrm{s}^{-1}$	$\dot{\gamma} = 6.310\mathrm{s}^{-1}$	$\dot{\gamma} = 10.000\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$\frac{[\eta]}{\mathrm{mPa}}$	387.0 ± 7.5	187.0 ± 24	260.5 ± 16	120.0 ± 23	37.0 ± 33	32.55 ± 34	0.95 ± 15	-27.8 ± 0.52	-29.5 ± 3.6	-28.96 ± 1.2	12.05 ± 14
$\frac{\eta_0}{\mathrm{mPa}}$	189.033 ± 0.97	39.83 ± 3.1	33.02 ± 2	23.43 ± 3	19.8 ± 4.2	9.34 ± 4.4	9.84 ± 2	11.53 ± 0.067	8.417 ± 0.46	6.162 ± 0.16	0.76 ± 1.8

Plot 16 Auftragung von η gegen c bei Saccharose für Scherrate 7



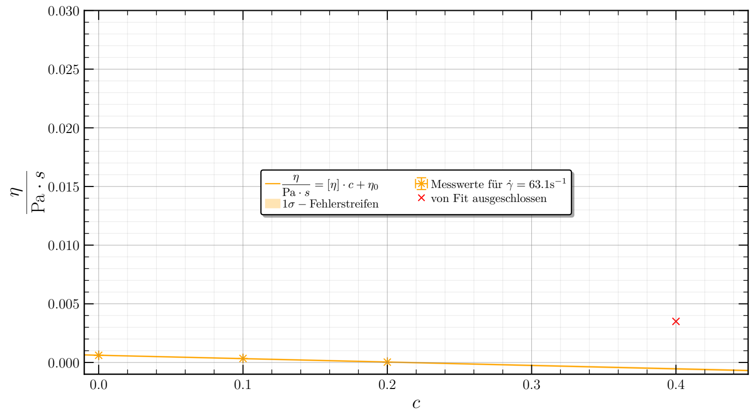
	$\dot{\gamma} = 1.000\mathrm{s}^{-1}$	$\dot{\gamma} = 1.585\mathrm{s}^{-1}$	$\dot{\gamma} = 2.511\mathrm{s}^{-1}$	$\dot{\gamma} = 3.981\mathrm{s}^{-1}$	$\dot{\gamma} = 6.310\mathrm{s}^{-1}$	$\dot{\gamma} = 10.000\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$\frac{[\eta]}{\mathrm{mPa}}$	387.0 ± 7.5	187.0 ± 24	260.5 ± 16	120.0 ± 23	37.0 ± 33	32.55 ± 34	0.95 ± 15	-27.8 ± 0.52	-29.5 ± 3.6	-28.96 ± 1.2	12.05 ± 14
$\frac{\eta_0}{\mathrm{mPa}}$	189.033 ± 0.97	39.83 ± 3.1	33.02 ± 2	23.43 ± 3	19.8 ± 4.2	9.34 ± 4.4	9.84 ± 2	11.53 ± 0.067	8.417 ± 0.46	6.162 ± 0.16	0.76 ± 1.8

Plot 17 Auftragung von η gegen c bei Saccharose für Scherrate 8



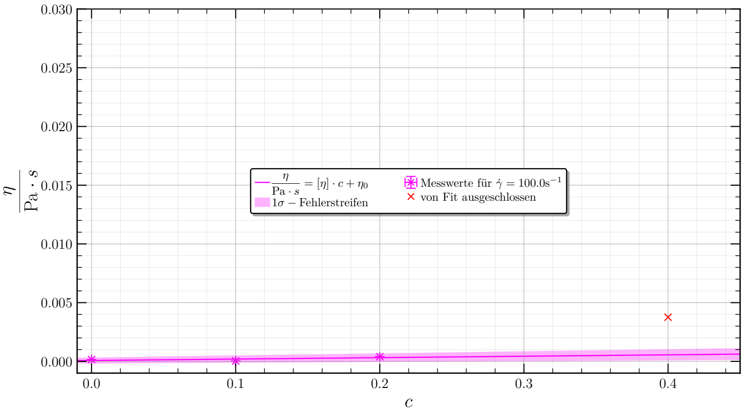
	$\dot\gamma = 1.000\mathrm{s}^{-1}$	$\dot\gamma = 1.585\mathrm{s}^{-1}$	$\dot\gamma = 2.511\mathrm{s}^{-1}$	$\dot\gamma = 3.981\mathrm{s}^{-1}$	$\dot\gamma = 6.310\mathrm{s}^{-1}$	$\dot\gamma = 10.000\mathrm{s}^{-1}$	$\dot\gamma = 15.850\mathrm{s}^{-1}$	$\dot\gamma = 25.120\mathrm{s}^{-1}$	$\dot\gamma = 39.810\mathrm{s}^{-1}$	$\dot\gamma = 63.100\mathrm{s}^{-1}$	$\dot\gamma = 100.000\mathrm{s}^{-1}$
$\frac{[\eta]}{\mathrm{mPa}}$	387.0 ± 7.5	187.0 ± 24	260.5 ± 16	120.0 ± 23	37.0 ± 33	32.55 ± 34	0.95 ± 15	-27.8 ± 0.52	-29.5 ± 3.6	-28.96 ± 1.2	12.05 ± 14
$\frac{\eta_0}{\mathrm{mPa}}$	189.033 ± 0.97	39.83 ± 3.1	33.02 ± 2	23.43 ± 3	19.8 ± 4.2	9.34 ± 4.4	9.84 ± 2	11.53 ± 0.067	8.417 ± 0.46	6.162 ± 0.16	0.76 ± 1.8

Plot 18 Auftragung von η gegen c bei Saccharose für Scherrate 9



	$\dot\gamma = 1.000\mathrm{s}^{-1}$	$\dot\gamma = 1.585\mathrm{s}^{-1}$	$\dot\gamma = 2.511\mathrm{s}^{-1}$	$\dot\gamma = 3.981\mathrm{s}^{-1}$	$\dot\gamma = 6.310\mathrm{s}^{-1}$	$\dot\gamma = 10.000\mathrm{s}^{-1}$	$\dot\gamma = 15.850\mathrm{s}^{-1}$	$\dot\gamma = 25.120\mathrm{s}^{-1}$	$\dot\gamma = 39.810\mathrm{s}^{-1}$	$\dot\gamma = 63.100\mathrm{s}^{-1}$	$\dot\gamma = 100.000\mathrm{s}^{-1}$
$\frac{[\eta]}{\mathrm{mPa}}$	387.0 ± 7.5	187.0 ± 24	260.5 ± 16	120.0 ± 23	37.0 ± 33	32.55 ± 34	0.95 ± 15	-27.8 ± 0.52	-29.5 ± 3.6	-28.96 ± 1.2	12.05 ± 14
$\frac{\eta_0}{\mathrm{mPa}}$	189.033 ± 0.97	39.83 ± 3.1	33.02 ± 2	23.43 ± 3	19.8 ± 4.2	9.34 ± 4.4	9.84 ± 2	11.53 ± 0.067	8.417 ± 0.46	6.162 ± 0.16	0.76 ± 1.8

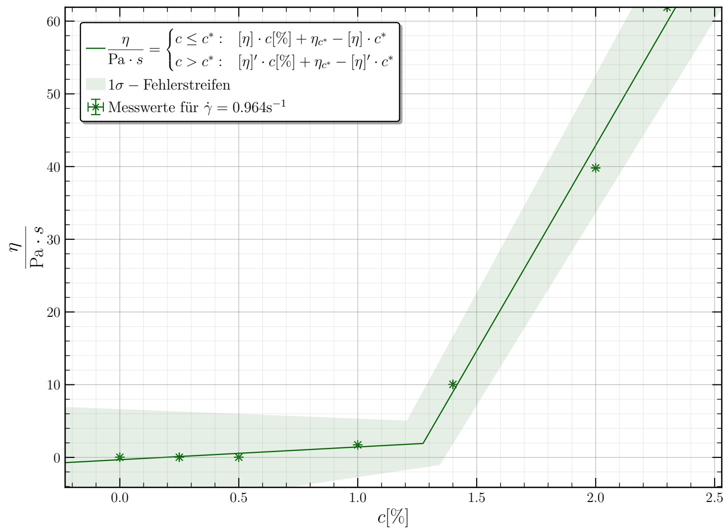
Plot 19 Auftragung von η gegen c bei Saccharose für Scherrate 10



	$\dot{\gamma} = 1.000\mathrm{s}^{-1}$	$\dot{\gamma} = 1.585\mathrm{s}^{-1}$	$\dot{\gamma} = 2.511\mathrm{s}^{-1}$	$\dot{\gamma} = 3.981\mathrm{s}^{-1}$	$\dot{\gamma} = 6.310\mathrm{s}^{-1}$	$\dot{\gamma} = 10.000\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$\frac{[\eta]}{\mathrm{mPa}}$	387.0 ± 7.5	187.0 ± 24	260.5 ± 16	120.0 ± 23	37.0 ± 33	32.55 ± 34	0.95 ± 15	-27.8 ± 0.52	-29.5 ± 3.6	-28.96 ± 1.2	12.05 ± 14
$\frac{\eta_0}{\mathrm{mPa}}$	189.033 ± 0.97	39.83 ± 3.1	33.02 ± 2	23.43 ± 3	19.8 ± 4.2	9.34 ± 4.4	9.84 ± 2	11.53 ± 0.067	8.417 ± 0.46	6.162 ± 0.16	0.76 ± 1.8

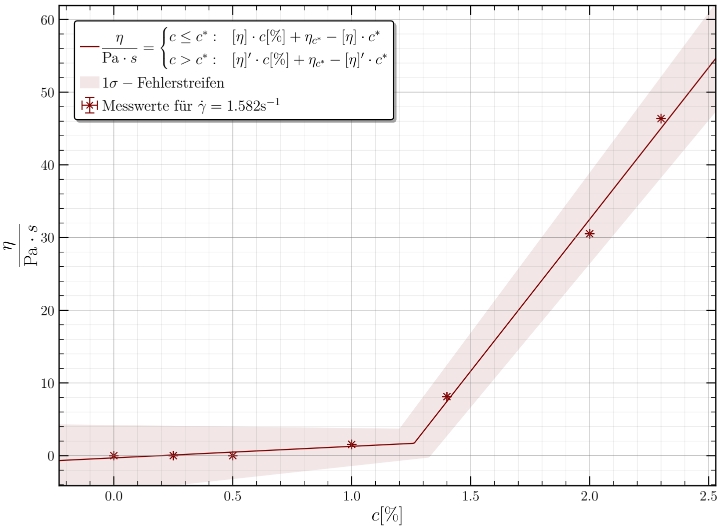
Plot 20 Auftragung von η gegen c bei Saccharose für Scherrate 11

A.2. Bestimmung der Überlappkonzentration von Guaran



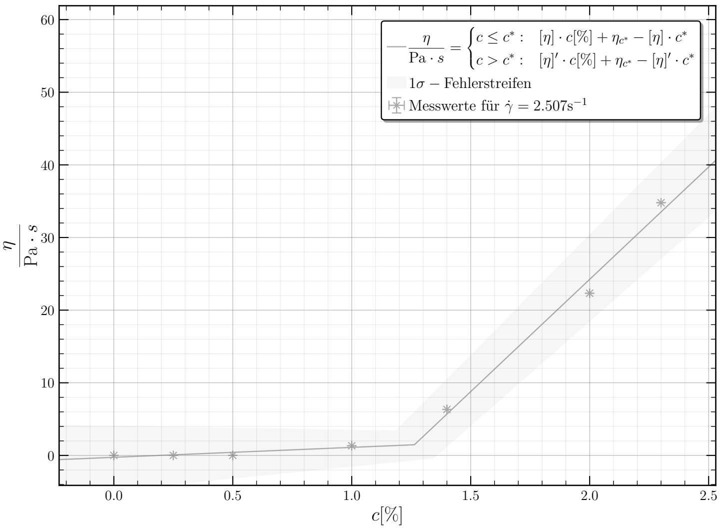
	$\dot{\gamma} = 0.964\mathrm{s}^{-1}$	$\dot{\gamma} = 1.582\mathrm{s}^{-1}$	$\dot{\gamma} = 2.507\mathrm{s}^{-1}$	$\dot{\gamma} = 3.979\mathrm{s}^{-1}$	$\dot{\gamma} = 6.304\mathrm{s}^{-1}$	$\dot{\gamma} = 9.999\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$c^*[\%]$	1.274 ± 0.07	1.262 ± 0.06	1.263 ± 0.08	1.257 ± 0.08	1.251 ± 0.08	1.245 ± 0.08	1.237 ± 0.08	1.23 ± 0.09	1.226 ± 0.09	1.216 ± 0.09	1.17 ± 0.1
$\frac{\eta_{c^*}}{\mathrm{Pa \cdot s}}$	1.91 ± 3	1.7 ± 2	1.46 ± 2	1.22 ± 1	0.98 ± 0.9	0.77 ± 0.6	0.59 ± 0.4	0.45 ± 0.3	0.34 ± 0.2	0.25 ± 0.1	0.182 ± 0.09
$\frac{[\eta]}{\mathrm{Pa \cdot s}}$	1.75 ± 3	1.59 ± 2	1.36 ± 2	1.15 ± 1	0.92 ± 1	0.73 ± 0.6	0.57 ± 0.4	0.43 ± 0.3	0.33 ± 0.2	0.24 ± 0.1	0.18 ± 0.09
$\frac{[\eta]'}{\mathrm{Pa \cdot s}}$	56.49 ± 3	41.75 ± 2	30.91 ± 2	21.79 ± 2	15.02 ± 1	10.17 ± 0.7	6.79 ± 0.5	4.5 ± 0.3	2.96 ± 0.2	1.92 ± 0.1	1.21 ± 0.1

Plot 21 Auftragung von η gegen c bei Guaran für Scherrate 1



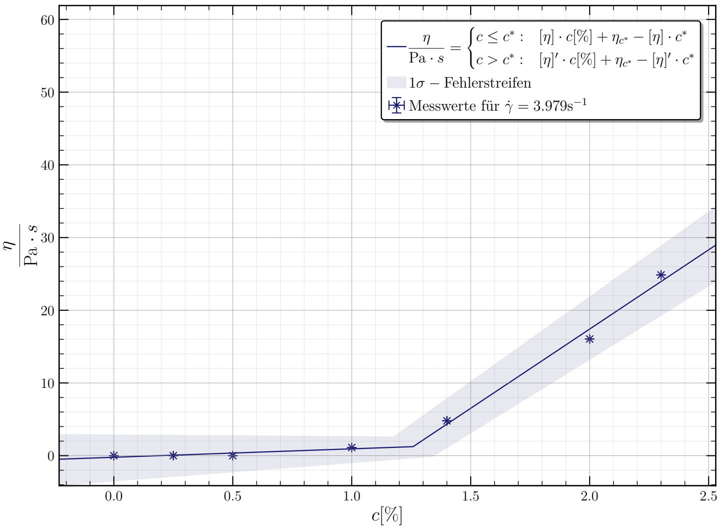
	$\dot{\gamma} = 0.964\mathrm{s}^{-1}$	$\dot{\gamma} = 1.582\mathrm{s}^{-1}$	$\dot{\gamma} = 2.507\mathrm{s}^{-1}$	$\dot{\gamma} = 3.979\mathrm{s}^{-1}$	$\dot{\gamma} = 6.304\mathrm{s}^{-1}$	$\dot{\gamma} = 9.999\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$c^*[\%]$	1.274 ± 0.07	1.262 ± 0.06	1.263 ± 0.08	1.257 ± 0.08	1.251 ± 0.08	1.245 ± 0.08	1.237 ± 0.08	1.23 ± 0.09	1.226 ± 0.09	1.216 ± 0.09	1.17 ± 0.1
$\frac{\eta_{c^*}}{\mathrm{Pa\cdot s}}$	1.91 ± 3	1.7 ± 2	1.46 ± 2	1.22 ± 1	0.98 ± 0.9	0.77 ± 0.6	0.59 ± 0.4	0.45 ± 0.3	0.34 ± 0.2	0.25 ± 0.1	0.182 ± 0.09
$\frac{[\eta]}{\mathrm{Pa\cdot s}}$	1.75 ± 3	1.59 ± 2	1.36 ± 2	1.15 ± 1	0.92 ± 1	0.73 ± 0.6	0.57 ± 0.4	0.43 ± 0.3	0.33 ± 0.2	0.24 ± 0.1	0.18 ± 0.09
$\frac{[\eta]'}{\mathrm{Pa\cdot s}}$	56.49 ± 3	41.75 ± 2	30.91 ± 2	21.79 ± 2	15.02 ± 1	10.17 ± 0.7	6.79 ± 0.5	4.5 ± 0.3	2.96 ± 0.2	1.92 ± 0.1	1.21 ± 0.1

Plot 22 Auftragung von η gegen c bei Guaran für Scherrate 2



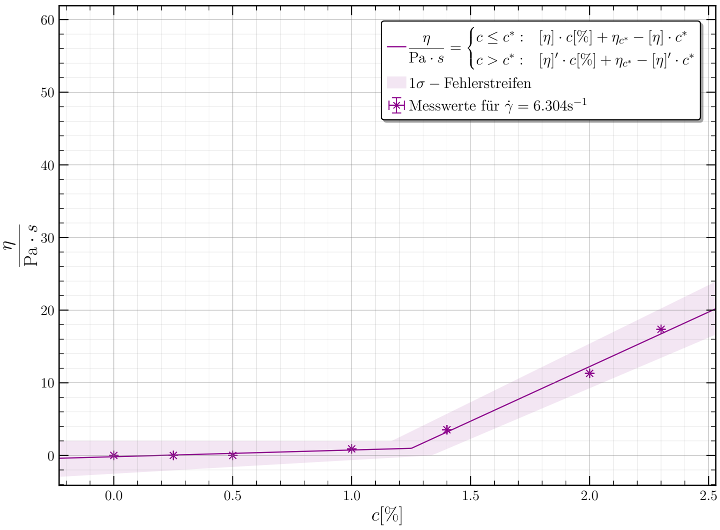
	$\dot{\gamma} = 0.964\mathrm{s}^{-1}$	$\dot{\gamma} = 1.582\mathrm{s}^{-1}$	$\dot{\gamma} = 2.507\mathrm{s}^{-1}$	$\dot{\gamma} = 3.979\mathrm{s}^{-1}$	$\dot{\gamma} = 6.304\mathrm{s}^{-1}$	$\dot{\gamma} = 9.999\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$c^*[\%]$	1.274 ± 0.07	1.262 ± 0.06	1.263 ± 0.08	1.257 ± 0.08	1.251 ± 0.08	1.245 ± 0.08	1.237 ± 0.08	1.23 ± 0.09	1.226 ± 0.09	1.216 ± 0.09	1.17 ± 0.1
$\frac{\eta_{c^*}}{\mathrm{Pa\cdot s}}$	1.91 ± 3	1.7 ± 2	1.46 ± 2	1.22 ± 1	0.98 ± 0.9	0.77 ± 0.6	0.59 ± 0.4	0.45 ± 0.3	0.34 ± 0.2	0.25 ± 0.1	0.182 ± 0.09
$\frac{[\eta]}{\mathrm{Pa\cdot s}}$	1.75 ± 3	1.59 ± 2	1.36 ± 2	1.15 ± 1	0.92 ± 1	0.73 ± 0.6	0.57 ± 0.4	0.43 ± 0.3	0.33 ± 0.2	0.24 ± 0.1	0.18 ± 0.09
$\frac{[\eta]'}{\mathrm{Pa\cdot s}}$	56.49 ± 3	41.75 ± 2	30.91 ± 2	21.79 ± 2	15.02 ± 1	10.17 ± 0.7	6.79 ± 0.5	4.5 ± 0.3	2.96 ± 0.2	1.92 ± 0.1	1.21 ± 0.1

Plot 23 Auftragung von η gegen c bei Guaran für Scherrate 3



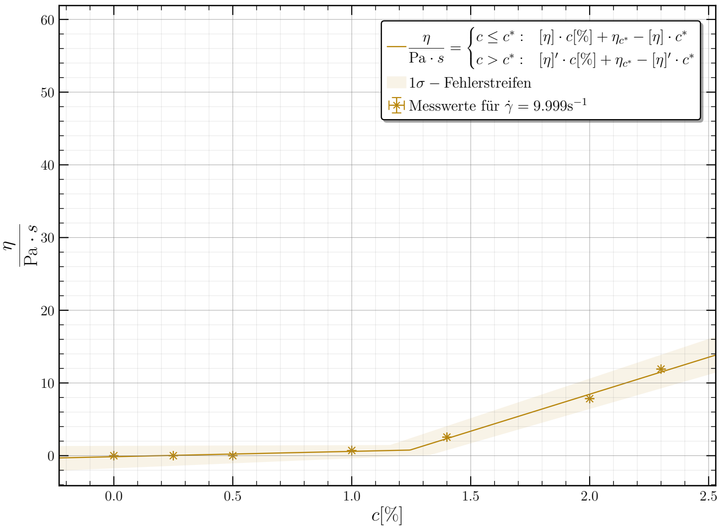
	$\dot{\gamma} = 0.964\mathrm{s}^{-1}$	$\dot{\gamma} = 1.582\mathrm{s}^{-1}$	$\dot{\gamma} = 2.507\mathrm{s}^{-1}$	$\dot{\gamma} = 3.979\mathrm{s}^{-1}$	$\dot{\gamma} = 6.304\mathrm{s}^{-1}$	$\dot{\gamma} = 9.999\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$c^*[\%]$	1.274 ± 0.07	1.262 ± 0.06	1.263 ± 0.08	1.257 ± 0.08	1.251 ± 0.08	1.245 ± 0.08	1.237 ± 0.08	1.23 ± 0.09	1.226 ± 0.09	1.216 ± 0.09	1.17 ± 0.1
$\frac{\eta_{c^*}}{\mathrm{Pa\cdot s}}$	1.91 ± 3	1.7 ± 2	1.46 ± 2	1.22 ± 1	0.98 ± 0.9	0.77 ± 0.6	0.59 ± 0.4	0.45 ± 0.3	0.34 ± 0.2	0.25 ± 0.1	0.182 ± 0.09
$\frac{[\eta]}{\mathrm{Pa\cdot s}}$	1.75 ± 3	1.59 ± 2	1.36 ± 2	1.15 ± 1	0.92 ± 1	0.73 ± 0.6	0.57 ± 0.4	0.43 ± 0.3	0.33 ± 0.2	0.24 ± 0.1	0.18 ± 0.09
$\frac{[\eta]'}{\mathrm{Pa\cdot s}}$	56.49 ± 3	41.75 ± 2	30.91 ± 2	21.79 ± 2	15.02 ± 1	10.17 ± 0.7	6.79 ± 0.5	4.5 ± 0.3	2.96 ± 0.2	1.92 ± 0.1	1.21 ± 0.1

Plot 24 Auftragung von η gegen c bei Guaran für Scherrate 4



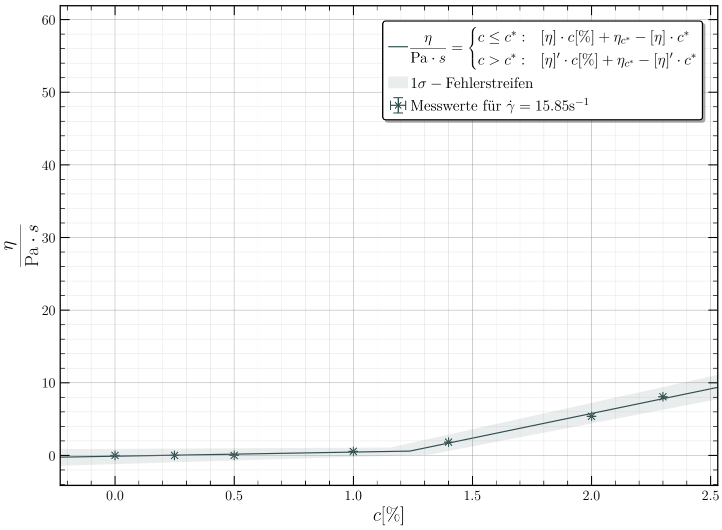
	$\dot{\gamma} = 0.964\mathrm{s}^{-1}$	$\dot{\gamma} = 1.582\mathrm{s}^{-1}$	$\dot{\gamma} = 2.507\mathrm{s}^{-1}$	$\dot{\gamma} = 3.979\mathrm{s}^{-1}$	$\dot{\gamma} = 6.304\mathrm{s}^{-1}$	$\dot{\gamma} = 9.999\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$c^*[\%]$	1.274 ± 0.07	1.262 ± 0.06	1.263 ± 0.08	1.257 ± 0.08	1.251 ± 0.08	1.245 ± 0.08	1.237 ± 0.08	1.23 ± 0.09	1.226 ± 0.09	1.216 ± 0.09	1.17 ± 0.1
$\frac{\eta_{c^*}}{\mathrm{Pa\cdot s}}$	1.91 ± 3	1.7 ± 2	1.46 ± 2	1.22 ± 1	0.98 ± 0.9	0.77 ± 0.6	0.59 ± 0.4	0.45 ± 0.3	0.34 ± 0.2	0.25 ± 0.1	0.182 ± 0.09
$\frac{[\eta]}{\mathrm{Pa\cdot s}}$	1.75 ± 3	1.59 ± 2	1.36 ± 2	1.15 ± 1	0.92 ± 1	0.73 ± 0.6	0.57 ± 0.4	0.43 ± 0.3	0.33 ± 0.2	0.24 ± 0.1	0.18 ± 0.09
$\frac{[\eta]'}{\mathrm{Pa\cdot s}}$	56.49 ± 3	41.75 ± 2	30.91 ± 2	21.79 ± 2	15.02 ± 1	10.17 ± 0.7	6.79 ± 0.5	4.5 ± 0.3	2.96 ± 0.2	1.92 ± 0.1	1.21 ± 0.1

Plot 25 Auftragung von η gegen c bei Guaran für Scherrate 5



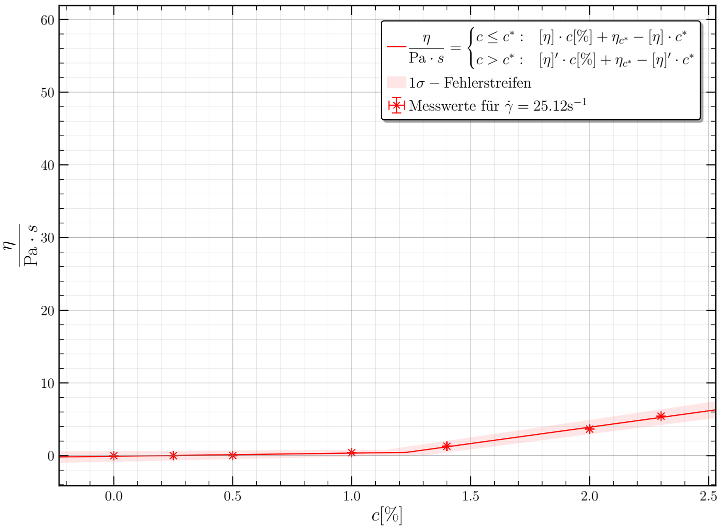
	$\dot{\gamma} = 0.964\mathrm{s}^{-1}$	$\dot{\gamma} = 1.582\mathrm{s}^{-1}$	$\dot{\gamma} = 2.507\mathrm{s}^{-1}$	$\dot{\gamma} = 3.979\mathrm{s}^{-1}$	$\dot{\gamma} = 6.304\mathrm{s}^{-1}$	$\dot{\gamma} = 9.999\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$c^*[\%]$	1.274 ± 0.07	1.262 ± 0.06	1.263 ± 0.08	1.257 ± 0.08	1.251 ± 0.08	1.245 ± 0.08	1.237 ± 0.08	1.23 ± 0.09	1.226 ± 0.09	1.216 ± 0.09	1.17 ± 0.1
$\frac{\eta_{c^*}}{\mathrm{Pa\cdot s}}$	1.91 ± 3	1.7 ± 2	1.46 ± 2	1.22 ± 1	0.98 ± 0.9	0.77 ± 0.6	0.59 ± 0.4	0.45 ± 0.3	0.34 ± 0.2	0.25 ± 0.1	0.182 ± 0.09
$\frac{[\eta]}{\mathrm{Pa\cdot s}}$	1.75 ± 3	1.59 ± 2	1.36 ± 2	1.15 ± 1	0.92 ± 1	0.73 ± 0.6	0.57 ± 0.4	0.43 ± 0.3	0.33 ± 0.2	0.24 ± 0.1	0.18 ± 0.09
$\frac{[\eta]'}{\mathrm{Pa\cdot s}}$	56.49 ± 3	41.75 ± 2	30.91 ± 2	21.79 ± 2	15.02 ± 1	10.17 ± 0.7	6.79 ± 0.5	4.5 ± 0.3	2.96 ± 0.2	1.92 ± 0.1	1.21 ± 0.1

Plot 26 Auftragung von η gegen c bei Guaran für Scherrate 6



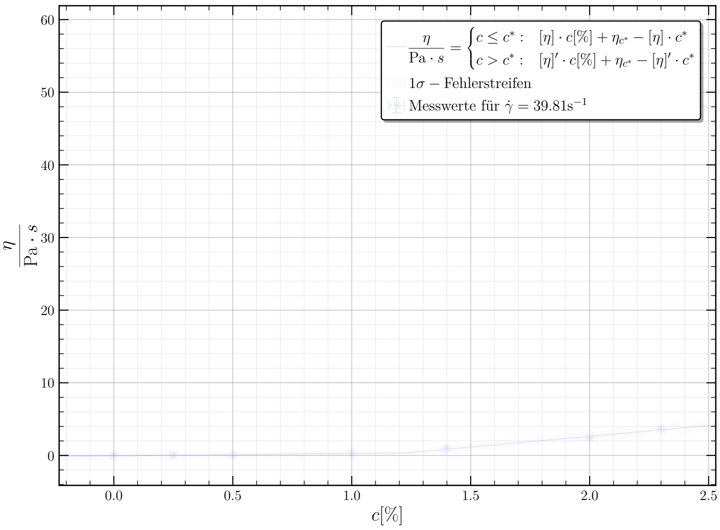
	$\dot{\gamma} = 0.964\mathrm{s}^{-1}$	$\dot{\gamma} = 1.582\mathrm{s}^{-1}$	$\dot{\gamma} = 2.507\mathrm{s}^{-1}$	$\dot{\gamma} = 3.979\mathrm{s}^{-1}$	$\dot{\gamma} = 6.304\mathrm{s}^{-1}$	$\dot{\gamma} = 9.999\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$c^*[\%]$	1.274 ± 0.07	1.262 ± 0.06	1.263 ± 0.08	1.257 ± 0.08	1.251 ± 0.08	1.245 ± 0.08	1.237 ± 0.08	1.23 ± 0.09	1.226 ± 0.09	1.216 ± 0.09	1.17 ± 0.1
$\frac{\eta_{c^*}}{\mathrm{Pa\cdot s}}$	1.91 ± 3	1.7 ± 2	1.46 ± 2	1.22 ± 1	0.98 ± 0.9	0.77 ± 0.6	0.59 ± 0.4	0.45 ± 0.3	0.34 ± 0.2	0.25 ± 0.1	0.182 ± 0.09
$\frac{[\eta]}{\mathrm{Pa\cdot s}}$	1.75 ± 3	1.59 ± 2	1.36 ± 2	1.15 ± 1	0.92 ± 1	0.73 ± 0.6	0.57 ± 0.4	0.43 ± 0.3	0.33 ± 0.2	0.24 ± 0.1	0.18 ± 0.09
$\frac{[\eta]'}{\mathrm{Pa\cdot s}}$	56.49 ± 3	41.75 ± 2	30.91 ± 2	21.79 ± 2	15.02 ± 1	10.17 ± 0.7	6.79 ± 0.5	4.5 ± 0.3	2.96 ± 0.2	1.92 ± 0.1	1.21 ± 0.1

Plot 27 Auftragung von η gegen c bei Guaran für Scherrate 7



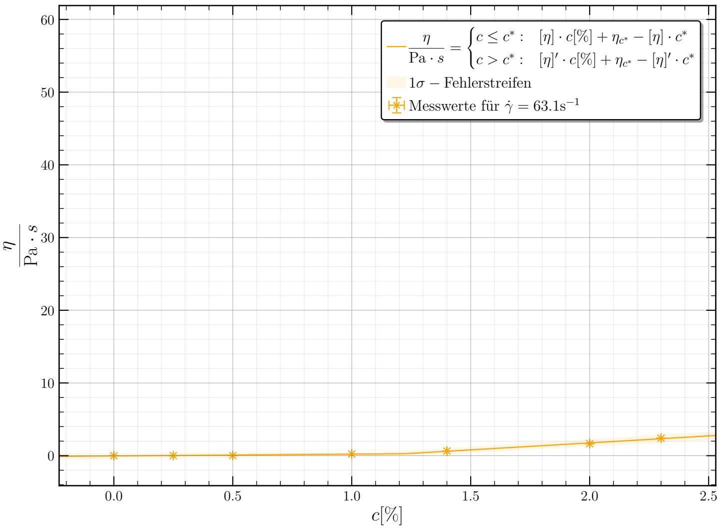
	$\dot{\gamma} = 0.964\mathrm{s}^{-1}$	$\dot{\gamma} = 1.582\mathrm{s}^{-1}$	$\dot{\gamma} = 2.507\mathrm{s}^{-1}$	$\dot{\gamma} = 3.979\mathrm{s}^{-1}$	$\dot{\gamma} = 6.304\mathrm{s}^{-1}$	$\dot{\gamma} = 9.999\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$c^*[\%]$	1.274 ± 0.07	1.262 ± 0.06	1.263 ± 0.08	1.257 ± 0.08	1.251 ± 0.08	1.245 ± 0.08	1.237 ± 0.08	1.23 ± 0.09	1.226 ± 0.09	1.216 ± 0.09	1.17 ± 0.1
$\frac{\eta_{c^*}}{\mathrm{Pa}\cdot s}$	1.91 ± 3	1.7 ± 2	1.46 ± 2	1.22 ± 1	0.98 ± 0.9	0.77 ± 0.6	0.59 ± 0.4	0.45 ± 0.3	0.34 ± 0.2	0.25 ± 0.1	0.182 ± 0.09
$\frac{[\eta]}{\mathrm{Pa}\cdot s}$	1.75 ± 3	1.59 ± 2	1.36 ± 2	1.15 ± 1	0.92 ± 1	0.73 ± 0.6	0.57 ± 0.4	0.43 ± 0.3	0.33 ± 0.2	0.24 ± 0.1	0.18 ± 0.09
$\frac{[\eta]'}{\mathrm{Pa}\cdot s}$	56.49 ± 3	41.75 ± 2	30.91 ± 2	21.79 ± 2	15.02 ± 1	10.17 ± 0.7	6.79 ± 0.5	4.5 ± 0.3	2.96 ± 0.2	1.92 ± 0.1	1.21 ± 0.1

Plot 28 Auftragung von η gegen c bei Guaran für Scherrate 8



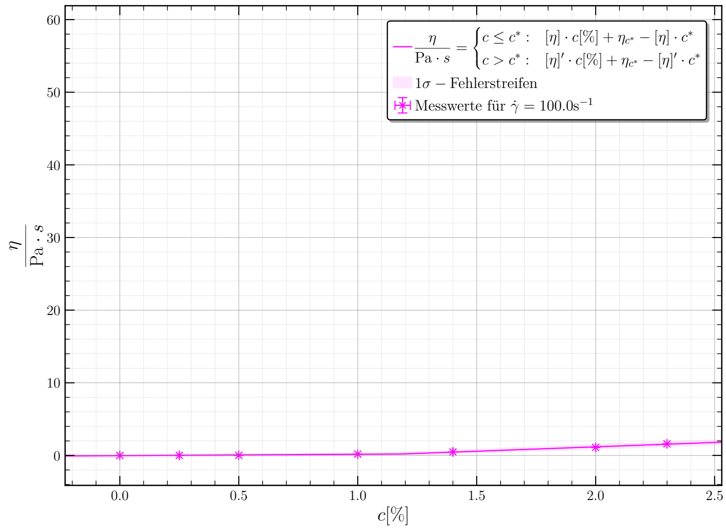
	$\dot{\gamma} = 0.964\mathrm{s}^{-1}$	$\dot{\gamma} = 1.582\mathrm{s}^{-1}$	$\dot{\gamma} = 2.507\mathrm{s}^{-1}$	$\dot{\gamma} = 3.979\mathrm{s}^{-1}$	$\dot{\gamma} = 6.304\mathrm{s}^{-1}$	$\dot{\gamma} = 9.999\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$c^*[\%]$	1.274 ± 0.07	1.262 ± 0.06	1.263 ± 0.08	1.257 ± 0.08	1.251 ± 0.08	1.245 ± 0.08	1.237 ± 0.08	1.23 ± 0.09	1.226 ± 0.09	1.216 ± 0.09	1.17 ± 0.1
$\frac{\eta_{c^*}}{\mathrm{Pa}\cdot s}$	1.91 ± 3	1.7 ± 2	1.46 ± 2	1.22 ± 1	0.98 ± 0.9	0.77 ± 0.6	0.59 ± 0.4	0.45 ± 0.3	0.34 ± 0.2	0.25 ± 0.1	0.182 ± 0.09
$\frac{[\eta]}{\mathrm{Pa}\cdot s}$	1.75 ± 3	1.59 ± 2	1.36 ± 2	1.15 ± 1	0.92 ± 1	0.73 ± 0.6	0.57 ± 0.4	0.43 ± 0.3	0.33 ± 0.2	0.24 ± 0.1	0.18 ± 0.09
$\frac{[\eta]'}{\mathrm{Pa}\cdot s}$	56.49 ± 3	41.75 ± 2	30.91 ± 2	21.79 ± 2	15.02 ± 1	10.17 ± 0.7	6.79 ± 0.5	4.5 ± 0.3	2.96 ± 0.2	1.92 ± 0.1	1.21 ± 0.1

Plot 29 Auftragung von η gegen c bei Guaran für Scherrate 9



	$\dot{\gamma} = 0.964\mathrm{s}^{-1}$	$\dot{\gamma} = 1.582\mathrm{s}^{-1}$	$\dot{\gamma} = 2.507\mathrm{s}^{-1}$	$\dot{\gamma} = 3.979\mathrm{s}^{-1}$	$\dot{\gamma} = 6.304\mathrm{s}^{-1}$	$\dot{\gamma} = 9.999\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$c^*[\%]$	1.274 ± 0.07	1.262 ± 0.06	1.263 ± 0.08	1.257 ± 0.08	1.251 ± 0.08	1.245 ± 0.08	1.237 ± 0.08	1.23 ± 0.09	1.226 ± 0.09	1.216 ± 0.09	1.17 ± 0.1
$\frac{\eta_{c^*}}{\mathrm{Pa}\cdot\mathrm{s}}$	1.91 ± 3	1.7 ± 2	1.46 ± 2	1.22 ± 1	0.98 ± 0.9	0.77 ± 0.6	0.59 ± 0.4	0.45 ± 0.3	0.34 ± 0.2	0.25 ± 0.1	0.182 ± 0.09
$\frac{[\eta]}{\mathrm{Pa}\cdot\mathrm{s}}$	1.75 ± 3	1.59 ± 2	1.36 ± 2	1.15 ± 1	0.92 ± 1	0.73 ± 0.6	0.57 ± 0.4	0.43 ± 0.3	0.33 ± 0.2	0.24 ± 0.1	0.18 ± 0.09
$\frac{[\eta]'}{\mathrm{Pa}\cdot\mathrm{s}}$	56.49 ± 3	41.75 ± 2	30.91 ± 2	21.79 ± 2	15.02 ± 1	10.17 ± 0.7	6.79 ± 0.5	4.5 ± 0.3	2.96 ± 0.2	1.92 ± 0.1	1.21 ± 0.1

Plot 30 Auftragung von η gegen c bei Guaran für Scherrate 10



	$\dot{\gamma} = 0.964\mathrm{s}^{-1}$	$\dot{\gamma} = 1.582\mathrm{s}^{-1}$	$\dot{\gamma} = 2.507\mathrm{s}^{-1}$	$\dot{\gamma} = 3.979\mathrm{s}^{-1}$	$\dot{\gamma} = 6.304\mathrm{s}^{-1}$	$\dot{\gamma} = 9.999\mathrm{s}^{-1}$	$\dot{\gamma} = 15.850\mathrm{s}^{-1}$	$\dot{\gamma} = 25.120\mathrm{s}^{-1}$	$\dot{\gamma} = 39.810\mathrm{s}^{-1}$	$\dot{\gamma} = 63.100\mathrm{s}^{-1}$	$\dot{\gamma} = 100.000\mathrm{s}^{-1}$
$c^*[\%]$	1.274 ± 0.07	1.262 ± 0.06	1.263 ± 0.08	1.257 ± 0.08	1.251 ± 0.08	1.245 ± 0.08	1.237 ± 0.08	1.23 ± 0.09	1.226 ± 0.09	1.216 ± 0.09	1.17 ± 0.1
$\frac{\eta_{c^*}}{\mathrm{Pa}\cdot\mathrm{s}}$	1.91 ± 3	1.7 ± 2	1.46 ± 2	1.22 ± 1	0.98 ± 0.9	0.77 ± 0.6	0.59 ± 0.4	0.45 ± 0.3	0.34 ± 0.2	0.25 ± 0.1	0.182 ± 0.09
$\frac{[\eta]}{\mathrm{Pa}\cdot\mathrm{s}}$	1.75 ± 3	1.59 ± 2	1.36 ± 2	1.15 ± 1	0.92 ± 1	0.73 ± 0.6	0.57 ± 0.4	0.43 ± 0.3	0.33 ± 0.2	0.24 ± 0.1	0.18 ± 0.09
$\frac{[\eta]'}{\mathrm{Pa}\cdot\mathrm{s}}$	56.49 ± 3	41.75 ± 2	30.91 ± 2	21.79 ± 2	15.02 ± 1	10.17 ± 0.7	6.79 ± 0.5	4.5 ± 0.3	2.96 ± 0.2	1.92 ± 0.1	1.21 ± 0.1

Plot 31 Auftragung von η gegen c bei Guaran für Scherrate 11

B. Python-Skripte zur Auswertung

B.1. Bestimmung des Potenzgesetzes für Saccharose

```python
# -*- coding: utf-8 -*-
"""
Created on Sat Dec  3 14:51:49 2022

@author: Jan-Philipp
"""

import numpy as np #math functions, arrays
import matplotlib.pyplot as plt #visualizing
from scipy.optimize import curve_fit
from itertools import product, combinations
import matplotlib
import pandas as pd

matplotlib.style.use('JaPh') #merely contains basic style information
plt.ioff()


def LinXY(x,y):
    """"Scale x and y to obtain an linear relation between ordinate and abscissa"""
    return np.log(x),np.log(y)

def LinXYerr(xerr,yerr,x,y):
    """"Scale xerr and yerr according to LinXY"""
    return xerr/np.abs(x),yerr/np.abs(y)

def LinRegr1(x,a,b):
    """Model of a linear function for fitting and plotting"""
    return a*x+b

def LinRegr(tup, x):
    """Model of a linear function for fitting and plotting"""
    a,b = tup
    return a*x+b

def LineIntersection(m1,m2,b1,b2):
    """determining the geometric center of the 1-sigma-range"""
    x = (b2-b1)/(m1-m2)
```

```
    y = m1*x+b1
    return x,y


def plot(CSVNAMES):

    fig,ax=plt.subplots(1,1,figsize=(10,10/np.sqrt(2)))

    colorlist = ['forestgreen','maroon','orange','darkgray','midnightblue','darkmagenta','darkgoldenrod','darkslategray','red','lavender','magenta']

    shear_ind = 2
    cs = [0,10,20,40]
    cs = np.array(cs)
    maxy = -np.inf
    miny = np.inf
    stdList = []
    poptList = []
    markers = ['o','x','.','^','.','+','d']
    ylabel = r'$\mathrm{ln}\left( \dfrac{\eta}{\mathrm{Pa}\cdot s}\right)$'
    xlabel = r'$\mathrm{ln}\left(\frac{\dot\gamma}{s^{-1}}\right)$'

    for i,CSVNAME in enumerate(CSVNAMES):
        PATH = CSVNAME + '.csv'
        df = pd.read_csv(PATH,sep=',',header=None)
        y=np.array(df.iloc[:,8])
        x=np.array(df.iloc[:,7])



        xerr_rel= xerr_abs= yerr_rel= yerr_abs = np.zeros(len(x))
        xerr = x*xerr_rel+xerr_abs
        yerr = y*yerr_rel+yerr_abs


        X,Y = LinXY(x,y)
        Xerr,Yerr = LinXYerr(xerr,yerr,x,y)

        maxy = max(Y) if maxy<max(Y) else maxy
        miny = min(Y) if miny>min(Y) else miny
```

```

        dx = (max(X)-min(X))/10
        dy = (maxy-miny)/15
        xlim = (min(X)-dx,max(X)+dx)
        ylim = (miny-dy,maxy+dy)


        popt, pcov = curve_fit(LinRegr1,X,Y,maxfev=10000)
        stdDev=np.sqrt(np.diag(pcov))

        t1 = np.linspace(xlim[0],xlim[1], 500)



        #ax.axvline(popt[0],marker='None',linestyle='dotted',color='navajowhite',label=r'$c^*$')
        #ax.axvspan(popt[0]-stdDev[0],popt[0]+stdDev[0],linestyle='None',alpha=.2,color='navajowhite')
        #ax.annotate(r'$c^*$',(popt[0],(popt[1]+min(Y))/2),xytext=((popt[0]*1.5+max(X)*.5)/(2),(popt[1]+min(Y))/2),arrowprops=dict(arrowstyle= matplotlib.patches.ArrowStyle("Fancy", head_length=.2, head_width=.2, tail_width=.1),color='black'))

        PointWise_LowerBound = np.zeros(len(t1))
        PointWise_UpperBound = np.zeros(len(t1))
        for tind in range(len(t1)):
            t = t1[tind]
            Min = LinRegr(popt,t)
            Max = LinRegr(popt,t)
            for comb1, comb2 in combinations(product([-1,1],repeat=len(stdDev)),2):
                p1 = popt+stdDev*comb1
                p2 = popt+stdDev*comb2
                Min = min(LinRegr(p1,t),Min)
                Max = max(LinRegr(p2,t),Max)
            PointWise_LowerBound[tind] = Min
            PointWise_UpperBound[tind] = Max

        if i == 0:
            ax.plot(t1,LinRegr(popt,t1), marker = 'None', linestyle = '-', color = colorlist[i], label=ylabel[:-1]+'=\\alpha \cdot'+xlabel[1:-1]+'+\\beta$')
```

```
            #ax.fill_between(t1,PointWise_LowerBound, PointWise_UpperBound, color=colorlist[i], alpha = .1, label = r'$1\sigma-\text{Fehlerstreifen}$')
        else:
            ax.plot(t1,LinRegr(popt,t1), marker = 'None', linestyle = '-', color = colorlist[i])
            #ax.fill_between(t1,PointWise_LowerBound, PointWise_UpperBound, color=colorlist[i], alpha = .1)
        ax.plot(X,Y,marker=markers[i], color = colorlist[i], linestyle='None',label=r'Messwerte für $c='+str(int(cs[i]))+'\%$')
        poptList.append(popt)
        stdList.append(stdDev)

    cell_text = np.core.defchararray.add(np.core.defchararray.add(np.round(np.transpose(poptList)).astype(str),np.full(np.transpose(poptList).shape,'+-')), np.round(np.transpose(stdList)).astype(str))
    #cell_text = np.core.defchararray.add(np.core.defchararray.add(np.full(np.full(np.transpose(poptList).shape,'$'),cell_text)),np.full(np.transpose(poptList).shape,'$'))
    sigfig = 2
    stdDevFlat = np.array(stdList).flatten()
    poptFlat = np.array(poptList).flatten()

    stdDev_rounded = ['{:g}'.format(float('{:.{p}g}'.format(stdDevFlat[i], p=sigfig)))+'$' for i in range(0,len(stdDevFlat))]
    decimals = [len(str(stdDev_rounded[i].split('.')[-1])) for i in range(0,len(stdDev_rounded))]
    popt_rounded = [r'$'+str(round(poptFlat[i],decimals[i]))+'\pm' for i in range(len(poptFlat))]
    poptList = np.reshape(np.array(popt_rounded),np.array(poptList).shape)
    stdList = np.reshape(np.array(stdDev_rounded),np.array(stdList).shape)
    cell_text = np.core.defchararray.add(np.transpose(poptList), np.transpose(stdList).astype(str))

    the_table = the_table = plt.table(cellText=cell_text,
                    #rowLabels = np.full(7,'a'),
                      rowLabels=[r'$\alpha$',r'$\beta$'],
```

```
                        colLabels=[r'$c='+str(i)+'\% $' for i in cs],
                        loc='bottom',
                        cellLoc='center',
                        bbox=[.1,-0.31,.8,0.2],
                        edges='closed')
    #the_table.auto_set_font_size(False)
    #the_table.set_fontsize(8)

    ax.set_ylim(ylim[0],maxy)
    ax.set_xlabel(xlabel,size=16)
    ax.set_ylabel(ylabel,size=16)
    ax.set_xlim(xlim[0],xlim[1])
    ax.grid(visible=True, which='minor', color="grey", linestyle='-',linewidth=.008, alpha=.2)

    ax.legend(loc='best',fontsize=13)


    #fig.tight_layout()

    #-----Save Figure------------
    plt.savefig("Water_Sucrose_Viscosity_vs_Shearrate_all_Concentrations.pdf",dpi=1200)

if __name__ == "__main__":
    plot(['sac_00_percent','sac_10_percent','sac_20_percent','sac_40_percent'])
```

```
# -*- coding: utf-8 -*-
"""
Created on Sat Dec  3 16:40:27 2022

@author: Jan-Philipp
"""


import numpy as np #math functions, arrays
import matplotlib.pyplot as plt #visualizing
from scipy.optimize import curve_fit
from itertools import product, combinations
import matplotlib
import pandas as pd
```

```
matplotlib.style.use('JaPh') #merely contains basic style information
plt.ioff()


def LinXY(x,y):
    """"Scale x and y to obtain an linear relation between ordinate and abscissa"""
    return np.log(x),np.log(y)

def LinXYerr(xerr,yerr,x,y):
    """"Scale xerr and yerr according to LinXY"""
    return xerr/np.abs(x),yerr/np.abs(y)

def LinRegr1(x,a,b):
    """Model of a linear function for fitting and plotting"""
    return a*x+b

def LinRegr(tup, x):
    """Model of a linear function for fitting and plotting"""
    a,b = tup
    return a*x+b

def LineIntersection(m1,m2,b1,b2):
    """determining the geometric center of the 1-sigma-range"""
    x = (b2-b1)/(m1-m2)
    y = m1*x+b1
    return x,y


def plot(CSVNAMES):

    fig,ax=plt.subplots(1,1,figsize=(10,10/np.sqrt(2)))

    colorlist = ['forestgreen','maroon','orange','darkgray','midnightblue','darkmagenta','darkgoldenrod','darkslategray','red','lavender','magenta']

    shear_ind = 2
    cs = [0,10,20,40]
    cs = np.array(cs)
    maxy = -np.inf
    miny = np.inf
```

```
    stdList = []
    poptList = []
    markers = ['o','x','.','^','.','+','d']
    ylabel = r'$\mathrm{ln}\left( \dfrac{\sigma}{\mathrm{Pa}}\right)$'
    xlabel = r'$\mathrm{ln}\left(\frac{\dot\gamma}{s^{-1}}\right)$'

    for i,CSVNAME in enumerate(CSVNAMES):
        PATH = CSVNAME + '.csv'
        df = pd.read_csv(PATH,sep=',',header=None)
        y=np.array(df.iloc[:,6])
        x=np.array(df.iloc[:,7])



        xerr_rel= xerr_abs= yerr_rel= yerr_abs = np.zeros(len(x))
        xerr = x*xerr_rel+xerr_abs
        yerr = y*yerr_rel+yerr_abs


        X,Y = LinXY(x,y)
        Xerr,Yerr = LinXYerr(xerr,yerr,x,y)

        maxy = max(Y) if maxy<max(Y) else maxy
        miny = min(Y) if miny>min(Y) else miny

        dx = (max(X)-min(X))/10
        dy = (maxy-miny)/15
        xlim = (min(X)-dx,max(X)+dx)
        ylim = (miny-dy,maxy+dy)


        popt, pcov = curve_fit(LinRegr1,X,Y,maxfev=10000)
        stdDev=np.sqrt(np.diag(pcov))

        t1 = np.linspace(xlim[0],xlim[1], 500)



        #ax.axvline(popt[0],marker='None',linestyle='dotted',color='navajowhite',label=r'$c^*$')
        #ax.axvspan(popt[0]-stdDev[0],popt[0]+stdDev[0],
```

```
            linestyle='None', alpha=.2, color='navajowhite')
        #ax.annotate(r'$c^*$',(popt[0],(popt[1]+min(Y))/2), xytext=((popt[0]*1.5+max(X)*.5)/(2),(popt[1]+min(Y))/2),arrowprops=dict(arrowstyle= matplotlib.patches.ArrowStyle("Fancy", head_length=.2, head_width=.2, tail_width=.1),color='black'))

        PointWise_LowerBound = np.zeros(len(t1))
        PointWise_UpperBound = np.zeros(len(t1))
        for tind in range(len(t1)):
            t = t1[tind]
            Min = LinRegr(popt,t)
            Max = LinRegr(popt,t)
            for comb1, comb2 in combinations(product([-1,1], repeat=len(stdDev)),2):
                p1 = popt+stdDev*comb1
                p2 = popt+stdDev*comb2
                Min = min(LinRegr(p1,t),Min)
                Max = max(LinRegr(p2,t),Max)
            PointWise_LowerBound[tind] = Min
            PointWise_UpperBound[tind] = Max

        if i == 0:
            ax.plot(t1,LinRegr(popt,t1), marker = 'None', linestyle = '-', color = colorlist[i], label= ylabel[:-1]+'=\\alpha \cdot'+xlabel[1:-1]+'+\\beta$')
            #ax.fill_between(t1,PointWise_LowerBound, PointWise_UpperBound,color=colorlist[i], alpha = .1, label = r'$1\sigma-\text{Fehlerstreifen}$')
        else:
            ax.plot(t1,LinRegr(popt,t1), marker = 'None', linestyle = '-', color = colorlist[i])
            #ax.fill_between(t1,PointWise_LowerBound, PointWise_UpperBound,color=colorlist[i], alpha = .1)
        ax.plot(X,Y,marker=markers[i], color = colorlist[i], linestyle='None',label=r'Messwerte für $c='+str(int(cs[i]))+'\%$')
        poptList.append(popt)
        stdList.append(stdDev)

    cell_text = np.core.defchararray.add(np.core.defchararray.add(np.round(np.transpose(poptList)).astype(str),np.full
```

```python
    (np.transpose(poptList).shape, '+-')), np.round(np.transpose(stdList)).astype(str))
    #cell_text = np.core.defchararray.add(np.core.defchararray.add(np.full(np.full(np.transpose(poptList).shape, '$'), cell_text)),np.full(np.transpose(poptList).shape, '$'))
    sigfig = 2
    stdDevFlat = np.array(stdList).flatten()
    poptFlat = np.array(poptList).flatten()

    stdDev_rounded = ['{:g}'.format(float('{:.{p}g}'.format(stdDevFlat[i], p=sigfig)))+'$' for i in range(0,len(stdDevFlat))]
    decimals = [len(str(stdDev_rounded[i].split('.')[-1])) for i in range(0,len(stdDev_rounded))]
    popt_rounded = [r'$'+str(round(poptFlat[i],decimals[i]))+'\pm' for i in range(len(poptFlat))]
    poptList = np.reshape(np.array(popt_rounded),np.array(poptList).shape)
    stdList = np.reshape(np.array(stdDev_rounded),np.array(stdList).shape)
    cell_text = np.core.defchararray.add(np.transpose(poptList), np.transpose(stdList).astype(str))

    the_table = the_table = plt.table(cellText=cell_text,
                        #rowLabels = np.full(7,'a'),
                          rowLabels=[r'$\alpha$',r'$\beta$'],
                          colLabels=[r'$c='+str(i)+'\% $' for i in cs],
                          loc='bottom',
                          cellLoc='center',
                          bbox=[.1,-0.31,.8,0.2],
                          edges='closed')
    #the_table.auto_set_font_size(False)
    #the_table.set_fontsize(8)

    ax.set_ylim(ylim[0],maxy)
    ax.set_xlabel(xlabel, size=16)
    ax.set_ylabel(ylabel, size=16)
    ax.set_xlim(xlim[0],xlim[1])
    ax.grid(visible=True, which='minor', color="grey", linestyle='-',linewidth=.008, alpha=.2)

    ax.legend(loc='best',fontsize=13)

```

```
    #fig.tight_layout()

    #-----Save Figure-------------
    plt.savefig("
        Water_Sucrose_Stress_vs_Shearrate_all_Concentrations.pdf
        ",dpi=1200)

if __name__ == "__main__":
    plot(['sac_00_percent','sac_10_percent','sac_20_percent','
        sac_40_percent'])
```

B.2. Bestimmung des Potenzgesetzes für die Messdaten aus der Literatur

```
# -*- coding: utf-8 -*-
"""
Created on Wed Dec 14 13:43:06 2022

@author: Jan-Philipp
"""


import numpy as np #math functions, arrays
import matplotlib.pyplot as plt #visualizing
from scipy.optimize import curve_fit
from itertools import product, combinations
import matplotlib
import pandas as pd

matplotlib.style.use('JaPh') #merely contains basic style
    information
plt.ioff()


def LinXY(x,y):
    """"Scale x and y to obtain an linear relation between
        ordinate and abscissa"""
    return x,y

def LinXYerr(xerr,yerr,x,y):
    """"Scale xerr and yerr according to LinXY"""
    return xerr,yerr
```

```python

def LinRegr1(x,a,b):
    """Model of a linear function for fitting and plotting"""
    return a*x+b

def LinRegr(tup, x):
    """Model of a linear function for fitting and plotting"""
    a,b = tup
    return a*x+b

def LineIntersection(m1,m2,b1,b2):
    """determining the geometric center of the 1-sigma-range"""
    x = (b2-b1)/(m1-m2)
    y = m1*x+b1
    return x,y


def plot():

    fig,ax=plt.subplots(1,1,figsize=(10,10/np.sqrt(2)))

    colorlist = ['forestgreen','maroon','orange','darkgray','midnightblue','darkmagenta','darkgoldenrod','darkslategray','red','lavender','magenta']

    shear_ind = 2
    cs = [1.45,1,.5,.25]
    cs = np.array(cs)
    maxy = -np.inf
    miny = np.inf
    stdList = []
    poptList = []
    markers = ['o','x','.','^','.','+','d']
    ylabel = r'$\mathrm{log}_{10}\left( \dfrac{\eta}{\mathrm{Pa}\cdot s}\right)$'
    xlabel = r'$\mathrm{log}_{10}\left(\frac{\dot\gamma}{s^{-1}}\right)$'

    imp = pd.read_csv('Guaran_Visk_gammadot_literatur.csv',sep=';',header=None)
    Xs = [np.array(imp.iloc[2*i,:]) for i in range(0,4)]
    Ys = [np.array(imp.iloc[2*i+1,:]) for i in range(0,4)]


```

```
    for i in range(0,len(Xs)):

        y=Ys[i]
        x=Xs[i]

        x = x[~np.isnan(x)]
        y = y[~np.isnan(y)]

        xerr_rel= xerr_abs= yerr_rel= yerr_abs = np.zeros(len(x))
        xerr = x*xerr_rel+xerr_abs
        yerr = y*yerr_rel+yerr_abs


        X,Y = LinXY(x,y)
        Xerr,Yerr = LinXYerr(xerr,yerr,x,y)

        maxy = max(Y) if maxy<max(Y) else maxy
        miny = min(Y) if miny>min(Y) else miny

        dx = (max(X)-min(X))/10
        dy = (maxy-miny)/15
        xlim = (min(X)-dx,max(X)+dx)
        ylim = (miny-dy,maxy+dy)


        popt, pcov = curve_fit(LinRegr1,X,Y,maxfev=10000)
        stdDev=np.sqrt(np.diag(pcov))

        t1 = np.linspace(xlim[0],xlim[1], 500)



        #ax.axvline(popt[0],marker='None',linestyle='dotted',color='navajowhite',label=r'$c^*$')
        #ax.axvspan(popt[0]-stdDev[0],popt[0]+stdDev[0],linestyle='None',alpha=.2,color='navajowhite')
        #ax.annotate(r'$c^*$',(popt[0],(popt[1]+min(Y))/2),xytext=((popt[0]*1.5+max(X)*.5)/(2),(popt[1]+min(Y))/2),arrowprops=dict(arrowstyle= matplotlib.patches.ArrowStyle("Fancy", head_length=.2, head_width=.2, tail_width=.1),color='black'))


```

```
        if i == 0:
            ax.plot(t1,LinRegr(popt,t1), marker = 'None', linestyle = '-', color = colorlist[i], label=ylabel[:-1]+'=\\alpha \cdot'+xlabel[1:-1]+'+\\beta$')
        else:
            ax.plot(t1,LinRegr(popt,t1), marker = 'None', linestyle = '-', color = colorlist[i])
        ax.plot(X,Y,marker=markers[i], color = colorlist[i], linestyle='None',label=r'Messwerte für $c='+str(np.round(cs[i],2))+'\%$')
        poptList.append(popt)
        stdList.append(stdDev)

    cell_text = np.core.defchararray.add(np.core.defchararray.add(np.round(np.transpose(poptList)).astype(str),np.full(np.transpose(poptList).shape,'+-')), np.round(np.transpose(stdList)).astype(str))
    #cell_text = np.core.defchararray.add(np.core.defchararray.add(np.full(np.full(np.transpose(poptList).shape,'$'), cell_text)),np.full(np.transpose(poptList).shape,'$'))
    sigfig = 2
    stdDevFlat = np.array(stdList).flatten()
    poptFlat = np.array(poptList).flatten()

    stdDev_rounded = ['{:g}'.format(float('{:.{p}g}'.format(stdDevFlat[i], p=sigfig)))+'$' for i in range(0,len(stdDevFlat))]
    decimals = [len(str(stdDev_rounded[i].split('.')[-1])) for i in range(0,len(stdDev_rounded))]
    popt_rounded = [r'$'+str(round(poptFlat[i],decimals[i]))+'\pm' for i in range(len(poptFlat))]
    poptList = np.reshape(np.array(popt_rounded),np.array(poptList).shape)
    stdList = np.reshape(np.array(stdDev_rounded),np.array(stdList).shape)
    cell_text = np.core.defchararray.add(np.transpose(poptList), np.transpose(stdList).astype(str))

    the_table = the_table = plt.table(cellText=cell_text,
                    #rowLabels = np.full(7,'a'),
                      rowLabels=[r'$\alpha$',r'$\beta$'],
                      colLabels=[r'$c='+str(np.round(i,2))+'\% $' for i in cs],
```

```
                             loc='bottom',
                             cellLoc='center',
                             bbox=[0,-0.31,1,0.2],
                             edges='closed')
    #the_table.auto_set_font_size(False)
    #the_table.set_fontsize(8)

    ax.set_ylim(ylim[0],maxy)
    ax.set_xlabel(xlabel, size=16)
    ax.set_ylabel(ylabel, size=16)
    ax.set_xlim(xlim[0], xlim[1])
    ax.grid(visible=True, which='minor', color="grey",
        linestyle='-',linewidth=.008, alpha=.2)

    ax.legend(loc='best',fontsize=11,ncol=2,columnspacing=0.6,
        labelspacing=.3)


    #fig.tight_layout()

    #---Save Figure-------
    plt.savefig("Guar_Viscosity_vs_Shearrate_literature.pdf",
        dpi=1200)


if __name__ == "__main__":
    plot()
```

B.3. Bestimmung des Potenzgesetzes für Guaran

```
# -*- coding: utf-8 -*-
"""
Created on Sat Dec  3 16:47:25 2022

@author: Jan-Philipp
"""

import numpy as np #math functions, arrays
import matplotlib.pyplot as plt #visualizing
from scipy.optimize import curve_fit
from itertools import product, combinations
import matplotlib
import pandas as pd
```

```

matplotlib.style.use('JaPh') #merely contains basic style information
plt.ioff()


def LinXY(x,y):
    """"Scale x and y to obtain an linear relation between ordinate and abscissa"""
    return np.log(x),np.log(y)

def LinXYerr(xerr,yerr,x,y):
    """"Scale xerr and yerr according to LinXY"""
    return xerr/np.abs(x),yerr/np.abs(y)

def LinRegr1(x,a,b):
    """Model of a linear function for fitting and plotting"""
    return a*x+b

def LinRegr(tup, x):
    """Model of a linear function for fitting and plotting"""
    a,b = tup
    return a*x+b

def LineIntersection(m1,m2,b1,b2):
    """determining the geometric center of the 1-sigma-range"""
    x = (b2-b1)/(m1-m2)
    y = m1*x+b1
    return x,y


def plot(CSVNAMES):

    fig,ax=plt.subplots(1,1,figsize=(10,10/np.sqrt(2)))

    colorlist = ['forestgreen','maroon','orange','darkgray','midnightblue','darkmagenta','darkgoldenrod','darkslategray','red','lavender','magenta']

    shear_ind = 2
    cs = [0,.25,.5,1,1.4,2,2.3]
    cs = np.array(cs)
    maxy = -np.inf
    miny = np.inf
```

```python
    stdList = []
    poptList = []
    markers = ['o','x','.','^','.','+','d']
    ylabel = r'$\mathrm{ln}\left( \dfrac{\sigma}{\mathrm{Pa}}\right)$'
    xlabel = r'$\mathrm{ln}\left(\frac{\dot\gamma}{s^{-1}}\right)$'

    for i,CSVNAME in enumerate(CSVNAMES):
        PATH = CSVNAME + '.csv'
        df = pd.read_csv(PATH, sep=',',header=None)
        y=np.array(df.iloc[:,6])
        x=np.array(df.iloc[:,7])



        xerr_rel= xerr_abs= yerr_rel= yerr_abs = np.zeros(len(x))
        xerr = x*xerr_rel+xerr_abs
        yerr = y*yerr_rel+yerr_abs


        X,Y = LinXY(x,y)
        Xerr,Yerr = LinXYerr(xerr,yerr,x,y)

        maxy = max(Y) if maxy<max(Y) else maxy
        miny = min(Y) if miny>min(Y) else miny

        dx = (max(X)-min(X))/10
        dy = (maxy-miny)/15
        xlim = (min(X)-dx,max(X)+dx)
        ylim = (miny-dy,maxy+dy)


        popt, pcov = curve_fit(LinRegr1,X,Y,maxfev=10000)
        stdDev=np.sqrt(np.diag(pcov))

        t1 = np.linspace(xlim[0],xlim[1], 500)



        #ax.axvline(popt[0],marker='None',linestyle='dotted',color='navajowhite',label=r'$c^*$')
```

```
        #ax.axvspan(popt[0]-stdDev[0],popt[0]+stdDev[0],linestyle='None',alpha=.2,color='navajowhite')
        #ax.annotate(r'$c^*$',(popt[0],(popt[1]+min(Y))/2),xytext=((popt[0]*1.5+max(X)*.5)/(2),(popt[1]+min(Y))/2),arrowprops=dict(arrowstyle= matplotlib.patches.ArrowStyle("Fancy", head_length=.2, head_width=.2, tail_width=.1),color='black'))
        """
        PointWise_LowerBound = np.zeros(len(t1))
        PointWise_UpperBound = np.zeros(len(t1))
        for tind in range(len(t1)):
            t = t1[tind]
            Min = LinRegr(popt,t)
            Max = LinRegr(popt,t)
            for comb1, comb2 in combinations(product([-1,1],repeat=len(stdDev)),2):
                p1 = popt+stdDev*comb1
                p2 = popt+stdDev*comb2
                Min = min(LinRegr(p1,t),Min)
                Max = max(LinRegr(p2,t),Max)
            PointWise_LowerBound[tind] = Min
            PointWise_UpperBound[tind] = Max
        """
        if i == 0:
            ax.plot(t1,LinRegr(popt,t1), marker = 'None', linestyle = '-', color = colorlist[i], label=ylabel[:-1]+'=\\alpha \cdot'+xlabel[1:-1]+'+\\beta$')
            #ax.fill_between(t1,PointWise_LowerBound,PointWise_UpperBound,color=colorlist[i], alpha = .1, label = r'$1\sigma-\text{Fehlerstreifen}$')
        else:
            ax.plot(t1,LinRegr(popt,t1), marker = 'None', linestyle = '-', color = colorlist[i])
            #ax.fill_between(t1,PointWise_LowerBound,PointWise_UpperBound,color=colorlist[i], alpha = .1)
        ax.plot(X,Y,marker=markers[i], color = colorlist[i], linestyle='None',label=r'Messwerte für $c='+str(np.round(cs[i],2))+'\%$')
        poptList.append(popt)
        stdList.append(stdDev)

```

```
    cell_text = np.core.defchararray.add(np.core.defchararray.add(np.round(np.transpose(poptList)).astype(str),np.full(np.transpose(poptList).shape,'+-')), np.round(np.transpose(stdList)).astype(str))
    #cell_text = np.core.defchararray.add(np.core.defchararray.add(np.full(np.full(np.transpose(poptList).shape,'$'),cell_text)),np.full(np.transpose(poptList).shape,'$'))
    sigfig = 2
    stdDevFlat = np.array(stdList).flatten()
    poptFlat = np.array(poptList).flatten()

    stdDev_rounded = ['{:g}'.format(float('{:.{p}g}'.format(stdDevFlat[i], p=sigfig)))+'$' for i in range(0,len(stdDevFlat))]
    decimals = [len(str(stdDev_rounded[i].split('.')[-1])) for i in range(0,len(stdDev_rounded))]
    popt_rounded = [r'$'+str(round(poptFlat[i],decimals[i]))+'\pm' for i in range(len(poptFlat))]
    poptList = np.reshape(np.array(popt_rounded),np.array(poptList).shape)
    stdList = np.reshape(np.array(stdDev_rounded),np.array(stdList).shape)
    cell_text = np.core.defchararray.add(np.transpose(poptList), np.transpose(stdList).astype(str))

    the_table = the_table = plt.table(cellText=cell_text,
                    #rowLabels = np.full(7,'a'),
                      rowLabels=[r'$\alpha$',r'$\beta$'],
                      colLabels=[r'$c='+str(np.round(i,2))+'\% $' for i in cs],
                      loc='bottom',
                      cellLoc='center',
                      bbox=[0,-0.31,1,0.2],
                      edges='closed')
    #the_table.auto_set_font_size(False)
    #the_table.set_fontsize(8)

    ax.set_ylim(ylim[0],maxy)
    ax.set_xlabel(xlabel,size=16)
    ax.set_ylabel(ylabel,size=16)
    ax.set_xlim(xlim[0],xlim[1])
    ax.grid(visible=True, which='minor', color="grey", linestyle='-',linewidth=.008, alpha=.2)

```

```
    ax.legend(loc='best',fontsize=11,ncol=2,columnspacing=0.6,labelspacing=.3)


    #fig.tight_layout()

    #-----Save Figure---------------
    plt.savefig("Guar_Stress_vs_Shearrate_all_Concentrations.pdf",dpi=1200)

if __name__ == "__main__":
    plot(['guar_00000_percent','guar_00025_percent','guar_00050_percent','guar_00100_percent','guar_00140_percent','guar_00200_percent','guar_00230_percent'])
```

```
# -*- coding: utf-8 -*-
"""
Created on Sat Dec  3 15:33:05 2022

@author: Jan-Philipp
"""

import numpy as np #math functions, arrays
import matplotlib.pyplot as plt #visualizing
from scipy.optimize import curve_fit
from itertools import product, combinations
import matplotlib
import pandas as pd

matplotlib.style.use('JaPh') #merely contains basic style information
plt.ioff()


def LinXY(x,y):
    """"Scale x and y to obtain an linear relation between ordinate and abscissa"""
    return np.log(x),np.log(y)

def LinXYerr(xerr,yerr,x,y):
    """"Scale xerr and yerr according to LinXY"""
    return xerr/np.abs(x),yerr/np.abs(y)

```

```
def LinRegr1(x,a,b):
    """Model of a linear function for fitting and plotting"""
    return a*x+b

def LinRegr(tup, x):
    """Model of a linear function for fitting and plotting"""
    a,b = tup
    return a*x+b

def LineIntersection(m1,m2,b1,b2):
    """determining the geometric center of the 1-sigma-range"""
    x = (b2-b1)/(m1-m2)
    y = m1*x+b1
    return x,y


def plot(CSVNAMES):

    fig,ax=plt.subplots(1,1,figsize=(10,10/np.sqrt(2)))

    colorlist = ['forestgreen','maroon','orange','darkgray','midnightblue','darkmagenta','darkgoldenrod','darkslategray','red','lavender','magenta']

    shear_ind = 2
    cs = [0,.25,.5,1,1.4,2,2.3]
    cs = np.array(cs)
    maxy = -np.inf
    miny = np.inf
    stdList = []
    poptList = []
    markers = ['o','x','.','^','.','+','d']
    ylabel = r'$\mathrm{ln}\left( \dfrac{\eta}{\mathrm{Pa}\cdot s}\right)$'
    xlabel = r'$\mathrm{ln}\left(\frac{\dot\gamma}{s^{-1}}\right)$'

    for i,CSVNAME in enumerate(CSVNAMES):
        PATH = CSVNAME + '.csv'
        df = pd.read_csv(PATH,sep=',',header=None)
        y=np.array(df.iloc[:,8])
        x=np.array(df.iloc[:,7])


```

```
        xerr_rel= xerr_abs= yerr_rel= yerr_abs = np.zeros(len(x))
        xerr = x*xerr_rel+xerr_abs
        yerr = y*yerr_rel+yerr_abs


        X,Y = LinXY(x,y)
        Xerr,Yerr = LinXYerr(xerr,yerr,x,y)

        maxy = max(Y) if maxy<max(Y) else maxy
        miny = min(Y) if miny>min(Y) else miny

        dx = (max(X)-min(X))/10
        dy = (maxy-miny)/15
        xlim = (min(X)-dx,max(X)+dx)
        ylim = (miny-dy,maxy+dy)


        popt, pcov = curve_fit(LinRegr1,X,Y,maxfev=10000)
        stdDev=np.sqrt(np.diag(pcov))

        t1 = np.linspace(xlim[0],xlim[1], 500)



        #ax.axvline(popt[0],marker='None',linestyle='dotted',color='navajowhite',label=r'$c^*$')
        #ax.axvspan(popt[0]-stdDev[0],popt[0]+stdDev[0],linestyle='None',alpha=.2,color='navajowhite')
        #ax.annotate(r'$c^*$',(popt[0],(popt[1]+min(Y))/2),xytext=((popt[0]*1.5+max(X)*.5)/(2),(popt[1]+min(Y))/2),arrowprops=dict(arrowstyle= matplotlib.patches.ArrowStyle("Fancy", head_length=.2, head_width=.2, tail_width=.1),color='black'))
        """
        PointWise_LowerBound = np.zeros(len(t1))
        PointWise_UpperBound = np.zeros(len(t1))
        for tind in range(len(t1)):
            t = t1[tind]
            Min = LinRegr(popt,t)
            Max = LinRegr(popt,t)
            for comb1, comb2 in combinations(product([-1,1], repeat=len(stdDev)),2):
```

```
                p1 = popt+stdDev*comb1
                p2 = popt+stdDev*comb2
                Min = min(LinRegr(p1,t),Min)
                Max = max(LinRegr(p2,t),Max)
            PointWise_LowerBound[tind] = Min
            PointWise_UpperBound[tind] = Max
        """
        if i == 0:
            ax.plot(t1,LinRegr(popt,t1), marker = 'None', linestyle = '-', color = colorlist[i], label=ylabel[:-1]+'=\\alpha \cdot'+xlabel[1:-1]+'+\\beta$')
            #ax.fill_between(t1,PointWise_LowerBound,PointWise_UpperBound,color=colorlist[i], alpha = .1, label = r'$1\sigma-\text{Fehlerstreifen}$')
        else:
            ax.plot(t1,LinRegr(popt,t1), marker = 'None', linestyle = '-', color = colorlist[i])
            #ax.fill_between(t1,PointWise_LowerBound,PointWise_UpperBound,color=colorlist[i], alpha = .1)
        ax.plot(X,Y,marker=markers[i], color = colorlist[i], linestyle='None',label=r'Messwerte für $c='+str(np.round(cs[i],2))+'\%$')
        poptList.append(popt)
        stdList.append(stdDev)

    cell_text = np.core.defchararray.add(np.core.defchararray.add(np.round(np.transpose(poptList)).astype(str),np.full(np.transpose(poptList).shape,'+-')), np.round(np.transpose(stdList)).astype(str))
    #cell_text = np.core.defchararray.add(np.core.defchararray.add(np.full(np.full(np.transpose(poptList).shape,'$'),cell_text)),np.full(np.transpose(poptList).shape,'$'))
    sigfig = 2
    stdDevFlat = np.array(stdList).flatten()
    poptFlat = np.array(poptList).flatten()

    stdDev_rounded = ['{:g}'.format(float('{:.{p}g}'.format(stdDevFlat[i], p=sigfig)))+'$' for i in range(0,len(stdDevFlat))]
    decimals = [len(str(stdDev_rounded[i].split('.')[-1])) for i in range(0,len(stdDev_rounded))]
    popt_rounded = [r'$'+str(round(poptFlat[i],decimals[i]))+'\
```

```
        pm' for i in range(len(poptFlat))]
    poptList = np.reshape(np.array(popt_rounded),np.array(
        poptList).shape)
    stdList = np.reshape(np.array(stdDev_rounded),np.array(
        stdList).shape)
    cell_text = np.core.defchararray.add(np.transpose(poptList)
        , np.transpose(stdList).astype(str))

    the_table = the_table = plt.table(cellText=cell_text ,
                    #rowLabels = np.full(7, 'a'),
                      rowLabels=[r'$\alpha$',r'$\beta$'],
                      colLabels=[r'$c='+str(np.round(i,2))+'\%␣
                         $' for i in cs],
                      loc='bottom',
                      cellLoc='center',
                      bbox=[0,-0.31,1,0.2],
                      edges='closed')
    #the_table.auto_set_font_size(False)
    #the_table.set_fontsize(8)

    ax.set_ylim(ylim[0],maxy)
    ax.set_xlabel(xlabel,size=16)
    ax.set_ylabel(ylabel,size=16)
    ax.set_xlim(xlim[0],xlim[1])
    ax.grid(visible=True, which='minor', color="grey",
        linestyle='-',linewidth=.008, alpha=.2)

    ax.legend(loc='best',fontsize=11,ncol=2,columnspacing=0.6,
        labelspacing=.3)


    #fig.tight_layout()

    #------Save Figure---------------
    plt.savefig("Guar_Viscosity_vs_Shearrate_all_Concentrations
        .pdf",dpi=1200)

if __name__ == "__main__":
    plot(['guar_00000_percent','guar_00025_percent','
        guar_00050_percent','guar_00100_percent','
        guar_00140_percent','guar_00200_percent','
        guar_00230_percent'])
```

B.4. Bestimmung der Konzentrationsabhängigkeit der Viskosität bei Saccharose

```
# -*- coding: utf-8 -*-
"""
Created on Sat Dec  3 16:40:27 2022

@author: Jan-Philipp
"""


import numpy as np #math functions, arrays
import matplotlib.pyplot as plt #visualizing
from scipy.optimize import curve_fit
from itertools import product, combinations
import matplotlib
import pandas as pd

matplotlib.style.use('JaPh') #merely contains basic style information
plt.ioff()


def LinXY(x,y):
    """"Scale x and y to obtain an linear relation between ordinate and abscissa"""
    return np.log(x),np.log(y)

def LinXYerr(xerr,yerr,x,y):
    """"Scale xerr and yerr according to LinXY"""
    return xerr/np.abs(x),yerr/np.abs(y)

def LinRegr1(x,a,b):
    """Model of a linear function for fitting and plotting"""
    return a*x+b

def LinRegr(tup, x):
    """Model of a linear function for fitting and plotting"""
    a,b = tup
    return a*x+b

def LineIntersection(m1,m2,b1,b2):
    """determining the geometric center of the 1-sigma-range"""
    x = (b2-b1)/(m1-m2)
```

```
    y = m1*x+b1
    return x,y


def plot(CSVNAMES):

    fig,ax=plt.subplots(1,1,figsize=(10,10/np.sqrt(2)))

    colorlist = ['forestgreen','maroon','orange','darkgray','midnightblue','darkmagenta','darkgoldenrod','darkslategray','red','lavender','magenta']

    shear_ind = 2
    cs = [0,10,20,40]
    cs = np.array(cs)
    maxy = -np.inf
    miny = np.inf
    stdList = []
    poptList = []
    markers = ['o','x','.','^','.','+','d']
    ylabel = r'$\mathrm{ln}\left( \dfrac{\sigma}{\mathrm{Pa}}\right)$'
    xlabel = r'$\mathrm{ln}\left(\frac{\dot\gamma}{s^{-1}}\right)$'

    for i,CSVNAME in enumerate(CSVNAMES):
        PATH = CSVNAME + '.csv'
        df = pd.read_csv(PATH,sep=',',header=None)
        y=np.array(df.iloc[:,6])
        x=np.array(df.iloc[:,7])



        xerr_rel= xerr_abs= yerr_rel= yerr_abs = np.zeros(len(x))
        xerr = x*xerr_rel+xerr_abs
        yerr = y*yerr_rel+yerr_abs


        X,Y = LinXY(x,y)
        Xerr,Yerr = LinXYerr(xerr,yerr,x,y)

        maxy = max(Y) if maxy<max(Y) else maxy
        miny = min(Y) if miny>min(Y) else miny
```

```
        dx = (max(X)-min(X))/10
        dy = (maxy-miny)/15
        xlim = (min(X)-dx,max(X)+dx)
        ylim = (miny-dy,maxy+dy)



        popt, pcov = curve_fit(LinRegr1,X,Y,maxfev=10000)
        stdDev=np.sqrt(np.diag(pcov))

        t1 = np.linspace(xlim[0],xlim[1], 500)



        #ax.axvline(popt[0],marker='None',linestyle='dotted',color='navajowhite',label=r'$c^*$')
        #ax.axvspan(popt[0]-stdDev[0],popt[0]+stdDev[0],linestyle='None',alpha=.2,color='navajowhite')
        #ax.annotate(r'$c^*$',(popt[0],(popt[1]+min(Y))/2),xytext=((popt[0]*1.5+max(X)*.5)/(2),(popt[1]+min(Y))/2),arrowprops=dict(arrowstyle= matplotlib.patches.ArrowStyle("Fancy", head_length=.2, head_width=.2, tail_width=.1),color='black'))

        PointWise_LowerBound = np.zeros(len(t1))
        PointWise_UpperBound = np.zeros(len(t1))
        for tind in range(len(t1)):
            t = t1[tind]
            Min = LinRegr(popt,t)
            Max = LinRegr(popt,t)
            for comb1, comb2 in combinations(product([-1,1],repeat=len(stdDev)),2):
                p1 = popt+stdDev*comb1
                p2 = popt+stdDev*comb2
                Min = min(LinRegr(p1,t),Min)
                Max = max(LinRegr(p2,t),Max)
            PointWise_LowerBound[tind] = Min
            PointWise_UpperBound[tind] = Max

        if i == 0:
            ax.plot(t1,LinRegr(popt,t1), marker = 'None', linestyle = '-', color = colorlist[i], label=ylabel[:-1]+'=\\alpha \cdot'+xlabel[1:-1]+'+\\beta$')
```

```
            #ax.fill_between(t1,PointWise_LowerBound,PointWise_UpperBound,color=colorlist[i], alpha = .1, label = r'$1\sigma-\text{Fehlerstreifen}$')
        else:
            ax.plot(t1,LinRegr(popt,t1), marker = 'None', linestyle = '-', color = colorlist[i])
            #ax.fill_between(t1,PointWise_LowerBound,PointWise_UpperBound,color=colorlist[i], alpha = .1)
        ax.plot(X,Y,marker=markers[i], color = colorlist[i], linestyle='None',label=r'Messwerte für $c='+str(int(cs[i]))+'\%$')
        poptList.append(popt)
        stdList.append(stdDev)

    cell_text = np.core.defchararray.add(np.core.defchararray.add(np.round(np.transpose(poptList)).astype(str),np.full(np.transpose(poptList).shape,'+-')), np.round(np.transpose(stdList)).astype(str))
    #cell_text = np.core.defchararray.add(np.core.defchararray.add(np.full(np.full(np.transpose(poptList).shape,'$'),cell_text)),np.full(np.transpose(poptList).shape,'$'))
    sigfig = 2
    stdDevFlat = np.array(stdList).flatten()
    poptFlat = np.array(poptList).flatten()

    stdDev_rounded = ['{:g}'.format(float('{:.{p}g}'.format(stdDevFlat[i], p=sigfig)))+'$' for i in range(0,len(stdDevFlat))]
    decimals = [len(str(stdDev_rounded[i].split('.')[-1])) for i in range(0,len(stdDev_rounded))]
    popt_rounded = [r'$'+str(round(poptFlat[i],decimals[i]))+'\pm' for i in range(len(poptFlat))]
    poptList = np.reshape(np.array(popt_rounded),np.array(poptList).shape)
    stdList = np.reshape(np.array(stdDev_rounded),np.array(stdList).shape)
    cell_text = np.core.defchararray.add(np.transpose(poptList), np.transpose(stdList).astype(str))

    the_table = the_table = plt.table(cellText=cell_text,
                      #rowLabels = np.full(7,'a'),
                        rowLabels=[r'$\alpha$',r'$\beta$'],
```

```
                        colLabels=[r'$c='+str(i)+'\%_$' for i in
                            cs],
                        loc='bottom',
                        cellLoc='center',
                        bbox=[.1,-0.31,.8,0.2],
                        edges='closed')
    #the_table.auto_set_font_size(False)
    #the_table.set_fontsize(8)

    ax.set_ylim(ylim[0],maxy)
    ax.set_xlabel(xlabel,size=16)
    ax.set_ylabel(ylabel,size=16)
    ax.set_xlim(xlim[0],xlim[1])
    ax.grid(visible=True, which='minor', color="grey",
        linestyle='-',linewidth=.008, alpha=.2)

    ax.legend(loc='best',fontsize=13)


    #fig.tight_layout()

    #------Save Figure-------------
    plt.savefig("
        Water_Sucrose_Stress_vs_Shearrate_all_Concentrations.pdf
        ",dpi=1200)

if __name__ == "__main__":
    plot(['sac_00_percent','sac_10_percent','sac_20_percent','
        sac_40_percent'])
```

B.5. Bestimmung der Konzentrationabhängigkeit der Viskosität bei Guaran

```
# -*- coding: utf-8 -*-
"""
Created on Mon Nov 28 10:49:25 2022

@author: Jan-Philipp
"""

import numpy as np #math functions, arrays
import matplotlib.pyplot as plt #visualizing
from scipy.optimize import curve_fit
```

```
from itertools import product, combinations
import matplotlib
import pandas as pd

matplotlib.style.use('JaPh') #merely contains basic style information
plt.ioff()


def LinXY(x,y):
    """"Scale x and y to obtain an linear relation between ordinate and abscissa"""
    return x,y


def LinXYerr(xerr,yerr,x,y):
    """"Scale xerr and yerr according to LinXY"""
    return xerr,yerr

def LinRegr1(x, x0, y0, k1, k2):
    """Model of a piecewise linear function for fitting and plotting"""
    tup = (x0, y0, k1, k2)
    return LinRegr(tup, x)

def LinRegr(tup, x):
    """Model of a piecewise linear function for fitting and plotting"""
    x0, y0, k1, k2 = tup
    return np.piecewise(x, [x < x0], [lambda x:k1*x + y0-k1*x0, lambda x:k2*x + y0-k2*x0])

def LineIntersection(m1,m2,b1,b2):
    """determining the geometric center of the 1-sigma-range"""
    x = (b2-b1)/(m1-m2)
    y = m1*x+b1
    return x,y

def plot(CSVNAMES):
    global cellcand

    fig,ax=plt.subplots(1,1,figsize=(10,10/np.sqrt(2)))

```

```
    colorlist = ['darkgreen','maroon','darkgray','midnightblue','darkmagenta','darkgoldenrod','darkslategray','red','lavender','orange','magenta']

    shear_ind = 2
    x = [0,.25,.5,1,1.4,2,2.3]
    x = np.array(x)
    maxy = 0
    stdList = []
    poptList = []
    gammadotList = []
    for shear_ind in range(0,11):
        y = []
        for CSVNAME in CSVNAMES:
            PATH = CSVNAME + '.csv'
            df = pd.read_csv(PATH, sep=',', header=None)
            y.append(df.iloc[shear_ind,8])

        y = np.array(y)

        maxy = max(y) if maxy<max(y) else maxy

        xerr_rel= xerr_abs= yerr_rel= yerr_abs = np.zeros(len(x))
        xerr = x*xerr_rel+xerr_abs
        yerr = y*yerr_rel+yerr_abs

        ylabel = r'$\dfrac{\eta}{\mathrm{Pa}\cdot s}$'
        xlabel = r'$c[\% ]$'

        X,Y = LinXY(x,y)
        Xerr,Yerr = LinXYerr(xerr,yerr,x,y)
        dx = (max(X)-min(X))/10
        dy = (maxy-min(Y))/15
        xlim = (min(X)-dx,max(X)+dx)
        ylim = (min(Y)-dy,maxy+dy)


        popt, pcov = curve_fit(LinRegr1,X,Y,p0=(1.3,5,2,35),maxfev=10000)
        stdDev=np.sqrt(np.diag(pcov))

        t1 = np.linspace(xlim[0],xlim[1], 500)

```

```python


        #ax.axvline(popt[0],marker='None',linestyle='dotted',color='navajowhite',label=r'$c^*$')
        #ax.axvspan(popt[0]-stdDev[0],popt[0]+stdDev[0],linestyle='None',alpha=.2,color='navajowhite')
        #ax.annotate(r'$c^*$',(popt[0],(popt[1]+min(Y))/2),xytext=((popt[0]*1.5+max(X)*.5)/(2),(popt[1]+min(Y))/2),arrowprops=dict(arrowstyle= matplotlib.patches.ArrowStyle("Fancy", head_length=.2, head_width=.2, tail_width=.1),color='black'))
        """
        PointWise_LowerBound = np.zeros(len(t1))
        PointWise_UpperBound = np.zeros(len(t1))
        for tind in range(len(t1)):
            t = t1[tind]
            Min = LinRegr(popt,t)
            Max = LinRegr(popt,t)
            for comb1, comb2 in combinations(product([-1,1], repeat=len(stdDev)),2):
                p1 = popt+stdDev*comb1
                p2 = popt+stdDev*comb2
                Min = min(LinRegr(p1,t),Min)
                Max = max(LinRegr(p2,t),Max)
            PointWise_LowerBound[tind] = Min
            PointWise_UpperBound[tind] = Max
        """
        if shear_ind == 0:
            ax.plot(t1,LinRegr(popt,t1), marker = 'None', linestyle = '-', color = colorlist[shear_ind], label=ylabel[:-1]+'=\\begin{cases} c\leq c^*: & [\eta ] \cdot '+xlabel[1:-1]+'+\eta_{c^*}-[\eta ]\cdot c^*  \\\ c> c^*: & [\eta ]^\prime \cdot '+xlabel[1:-1]+'+\eta_{c^*}-[\eta ]^\prime\cdot c^*  \end{cases}$')
            #ax.fill_between(t1,PointWise_LowerBound,PointWise_UpperBound,color=colorlist[shear_ind], alpha = .1, label = r'$1\sigma-\text{Fehlerstreifen}$')
        else:
            ax.plot(t1,LinRegr(popt,t1), marker = 'None', linestyle = '-', color = colorlist[shear_ind])
            #ax.fill_between(t1,PointWise_LowerBound,PointWise_UpperBound,color=colorlist[shear_ind],
```

```
                    alpha = .1)
            ax.errorbar(x=X,y=Y,xerr=Xerr,yerr=Yerr,marker='x',
                color = colorlist[shear_ind],linestyle='None',label=
                r'Messwerte für $\dot\gamma='+str(round(df.iloc[
                shear_ind,7],3))+'\mathrm s^{-1}$')
            poptList.append(popt)
            stdList.append(stdDev)
            gammadotList.append(df.iloc[shear_ind,7])

    cell_text = np.core.defchararray.add(np.core.defchararray.
        add(np.round(np.transpose(poptList)).astype(str),np.full
        (np.transpose(poptList).shape,'+-')), np.round(np.
        transpose(stdList)).astype(str))
    #cell_text = np.core.defchararray.add(np.core.defchararray.
        add(np.full(np.full(np.transpose(poptList).shape,'$'),
        cell_text)),np.full(np.transpose(poptList).shape,'$'))
    sigfig = 1
    stdDevFlat = np.array(stdList).flatten()
    poptFlat = np.array(poptList).flatten()

    np.savetxt('c_star_by_gamma_dot.csv',[np.transpose(poptList
        )[0],np.transpose(stdList)[0],gammadotList],delimiter=',
        ')

    stdDev_rounded = ['{:g}'.format(float('{:.{p}g}'.format(
        stdDevFlat[i], p=sigfig)))+'$' for i in range(0,len(
        stdDevFlat))]
    decimals = [len(str(stdDev_rounded[i].split('.')[-1])) for
        i in range(0,len(stdDev_rounded))]
    popt_rounded = [r'$'+str(round(poptFlat[i],decimals[i]))+'\
        pm' for i in range(len(poptFlat))]
    poptList = np.reshape(np.array(popt_rounded),np.array(
        poptList).shape)
    stdList = np.reshape(np.array(stdDev_rounded),np.array(
        stdList).shape)
    cell_text = np.core.defchararray.add(np.transpose(poptList)
        , np.transpose(stdList).astype(str))

    the_table = the_table = plt.table(cellText=cell_text,
                        #rowLabels = np.full(7,'a'),
                            rowLabels=[r'$c^*[\% ]$',r'$\frac{\eta_{c
                                ^*}}{\mathrm{Pa\cdot s}}$',r'$\frac{[\
                                eta ]}{\mathrm{Pa\cdot s}}$',r'$\frac
                                {[\eta ]^\prime}{\mathrm{Pa\cdot s}}$'
```

```
                            ],
                            colLabels=[r'$\dot\gamma=%.3f\mathrm_s
                                ^{-1}$'%i for i in gammadotList],#['
                                Wert', 'Unsicherheit'],
                            loc='bottom',
                            cellLoc='center',
                            bbox=[-0.1,-0.4,1.2,0.3],
                            edges='closed')
    the_table.auto_set_font_size(False)
    the_table.set_fontsize(8)

    ax.set_ylim(ylim[0],maxy)
    ax.set_xlabel(xlabel,size=16)
    ax.set_ylabel(ylabel,size=16)
    ax.set_xlim(xlim[0],xlim[1])
    ax.grid(visible=True, which='minor', color="grey",
        linestyle='-',linewidth=.008, alpha=.2)

    ax.legend(loc='best',fontsize=13)


    #fig.tight_layout()

    #----Save Figure-----------
    plt.savefig("
        Viscosity_by_Concentration_Guar_all_Shearrates_without_1sigma
        .pdf",dpi=1200)

if __name__ == "__main__":
    plot(['guar_00000_percent','guar_00025_percent','
        guar_00050_percent','guar_00100_percent','
        guar_00140_percent','guar_00200_percent','
        guar_00230_percent'])
```

B.6. Frequenzversuch

```
# -*- coding: utf-8 -*-
"""
Created on Sat Dec  3 11:27:48 2022

@author: Jan-Philipp
"""

```

```
import numpy as np #math functions, arrays
import matplotlib.pyplot as plt #visualizing
from scipy.interpolate import make_interp_spline
import matplotlib
import pandas as pd
from scipy.optimize import curve_fit

matplotlib.style.use('JaPh') #merely contains basic style information
plt.ioff()

def Fit(x,a,b,c,d):
    return np.tanh(a*(x-b))*c+d

def plot(CSVNAME):
    global popt1
    PATH = CSVNAME + '.csv'
    fig,ax=plt.subplots(1,1,figsize=(10,10/np.sqrt(2)))

    df = pd.read_csv(PATH,sep=',',header=None)
    Y1=df.iloc[:,10]
    Y2=df.iloc[:,11]
    X =df.iloc[:,6]
    print(Y1,Y2,X)
    X = np.log(X)
    Y1 = np.log(Y1)
    Y2 = np.log(Y2)

    ylabel = r'$\mathrm{ln}\left(\frac{G^\prime}{\mathrm{Pa}}\right), \mathrm{ln}\left(\frac{G^{\prime\prime}}{\mathrm{Pa}}\right)$'
    xlabel = r'$\mathrm{ln}\left(\frac{f}{\mathrm{Hz}}\right)$'

    Ymax = np.max(np.array([Y1,Y2]),axis=0)
    Ymin = np.min(np.array([Y1,Y2]),axis=0)

    dx = (max(X)-min(X))/10
    dy = (max(Ymax)-min(Ymin))/15
    xlim = (min(X)-dx,max(X)+dx)
    ylim = (min(Ymin)-dy,max(Ymax)+dy)

    #X_Y1_Spline = make_interp_spline(X, Y1,k=3)
    #X_Y2_Spline = make_interp_spline(X, Y2,k=3)

```

```
        popt1,pcov1 = curve_fit(Fit,X,Y1,maxfev = 100000,p0=[0.79769266, 3.77394899, 5.19260498, 7.45366197])
        popt2,pcov2 = curve_fit(Fit,X,Y2,maxfev = 100000,p0=[0.79769266, 3.77394899, 5.19260498, 7.45366197])

        t1 = np.linspace(min(X),max(X), 10**3)

        minind1 = np.argmin(np.round(np.abs(Fit(t1,*popt2)-Fit(t1,*popt1)),1))
        minind2 = np.argmin(np.round(np.abs(Fit(t1,*popt2)-Fit(t1,*popt1)),4))


        #ax.plot(t1,Fit(t1,*popt2),marker='None',linestyle='-',color='red')
        #ax.plot(t1,Fit(t1,*popt1),marker='None',linestyle='-',color='blue')
        #ax.plot(t1,X_Y1_Spline(t1),marker='None',color='black',linestyle='-')
        #ax.plot(t1,X_Y2_Spline(t1),marker='None',color='black',linestyle='-')
        ax.plot(X,Y2,marker='x',linestyle='None',color='red',label=r'$\mathrm{ln}\left(\frac{G^{{\prime\prime}}}{\mathrm{Pa}}\right)$')
        ax.plot(X,Y1,marker='x',linestyle='None',color='blue',label=r'$\mathrm{ln}\left(\frac{G^{\prime}}{\mathrm{Pa}}\right)$')


        #ax.axvline((t1[minind2]+t1[minind1])/2,marker='None',linestyle='dotted',color='navajowhite',label=r'Übergangsfrequenz $f^*=\left(%.2f\pm%.2f\right)\ \mathrm{Hz}$'%(np.exp((t1[minind2]+t1[minind1])/2),(np.exp(t1[minind2])-np.exp(t1[minind1]))/2))
        #ax.axvline(t1[minind2],marker='None',linestyle='dotted',color='navajowhite')
        #ax.axvspan(t1[minind1],t1[minind2],linestyle='None',alpha=.2,color='navajowhite',label=r'$\mathrm{ln}\left(\frac{G^{{\prime\prime}}}{\mathrm{Pa}}\right)\approx \mathrm{ln}\left(\frac{G^{\prime}}{\mathrm{Pa}}\right)$')
        ax.axvspan(-10,-1,linestyle='None',alpha=.2,color='lightgreen',label=r'$\left(\mathrm{ln}\left(\frac{G^{{\prime\prime}}}{\mathrm{Pa}}\right)- \mathrm{ln}\left(\
```

```
    frac{G^{\prime}}{\mathrm{Pa}}\right)\right)\approx \mathrm{const.}<0$')
    ax.axvspan(-1,(t1[minind2]+t1[minind1])/2,linestyle='None',alpha=.2,color='lightcoral',label=r'$\mathrm{ln}\left(\frac{G^{{\prime\prime}}}{\mathrm{Pa}}\right)< \mathrm{ln}\left(\frac{G^{\prime}}{\mathrm{Pa}}\right)$')
    ax.axvspan((t1[minind2]+t1[minind1])/2,4.3,linestyle='None',alpha=.2,color='lightskyblue',label=r'$\mathrm{ln}\left(\frac{G^{{\prime\prime}}}{\mathrm{Pa}}\right)> \mathrm{ln}\left(\frac{G^{\prime}}{\mathrm{Pa}}\right)$')
    ax.axvspan(4.3,10,linestyle='None',alpha=.2,color='midnightblue',label=r'$\mathrm{ln}\left(\frac{G^{{\prime\prime}}}{\mathrm{Pa}}\right)< \mathrm{ln}\left(\frac{G^{\prime}}{\mathrm{Pa}}\right)$')
    #ax.annotate(r'$c^*$',(popt[0],(popt[1]+min(Y))/2),xytext=((popt[0]*1.5+max(X)*.5)/(2),(popt[1]+min(Y))/2),arrowprops=dict(arrowstyle= matplotlib.patches.ArrowStyle("Fancy", head_length=.2, head_width=.2, tail_width=.1),color='black'))


    ax.set_ylim(ylim[0],ylim[1])
    ax.set_xlabel(xlabel,size=16)
    ax.set_ylabel(ylabel,size=16)
    ax.set_xlim(xlim[0],xlim[1])
    ax.grid(visible=True, which='minor', color="grey", linestyle='-',linewidth=.008, alpha=.2)

    ax.legend(loc='best',fontsize=15)
    fig.tight_layout()

    #-----Save Figure------------
    plt.savefig("Osc_Guaran.pdf",dpi=1200)

if __name__ == "__main__":
    plot('Osc_Guaran')
```