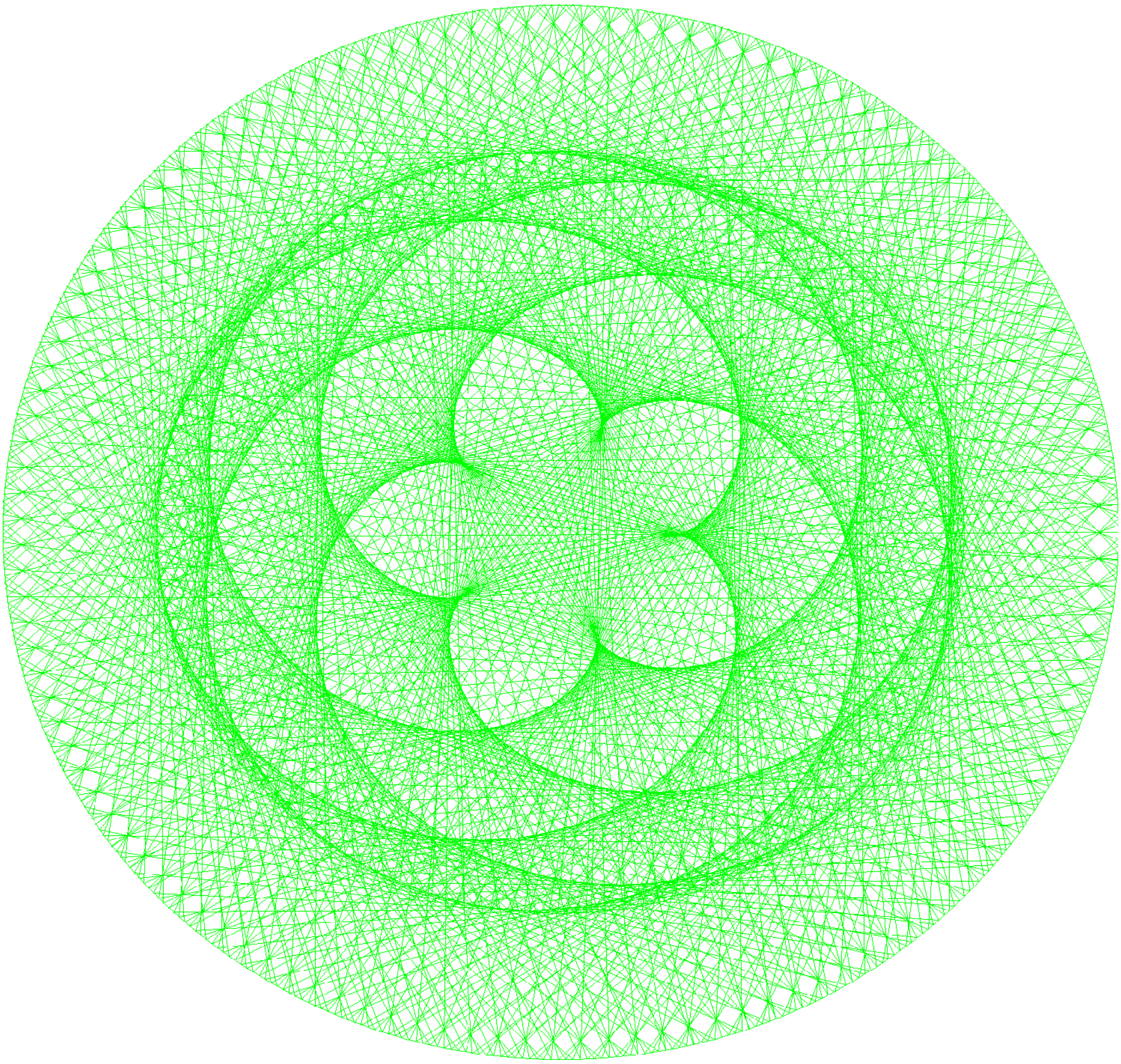
29.11.2019

Kepler's Sphärengeometrie

Zufall oder steckt dahinter eine höhere Ordnung?



Michael Voemel
Widnauerstrasse 1, 9435 Heerbrugg
Kantonsschule Heerbrugg, 4Pb
Betreuer: Herr Bernard Carlo

Inhaltsverzeichnis

1. Einleitung ... 3
2. Johannes Keplers Geschichte ... 3
3. Unser Sonnensystem ... 4
4. Gedanklicher Abriss der Vorgehensweise ... 5
5. Die Keplerschen Gesetze ... 6
5.1. Das 1. Keplersche Gesetz ... 6
5.2. Das 2. Keplersche Gesetz ... 6
5.3. Das 3. Keplersche Gesetz ... 6
6. Berechnung der Planetenbahnen ... 8
7. Planetenbahnen in Beziehung gesetzt ... 13
8. Programmierung in TigerJython ... 17
8.1. Das Vorprogramm ... 18
8.2. Die finale Software - Lösung ... 19
8.2.1. Datenbanken und Rohdaten ... 20
8.2.2. Der Konstruktor ... 21
8.2.3. Methoden ... 23
8.3. Programminterface ... 28
9. Formen aus Planetenkombinationen ... 28
9.1. Venus-Erde Figur ... 29
9.2. Erde-Jupiter Figur ... 30
9.3. Jupiter-Saturn Figur ... 31
9.4. Mars-Jupiter Figur ... 32
9.5. Erde-Mars Figur ... 33
9.6. Saturn-Uranus Figur ... 34
9.7. Merkur-Venus Figur ... 35
10. Schlussfolgerung/Resultat ... 36
11. Weiterführende Arbeiten und Ausblick ... 37
12. Zusammenfassung ... 38
13. Quellenverzeichnis ... 39
14. Literaturverzeichnis ... 39
15. Abbildungsverzeichnis ... 40
16. Tabellenverzeichnis ... 41
17. Nachwort und Danksagung ... 42
18. Anhang ... 43

18.1. Vorprogramm ... 43
18.2. Programm ... 44
18.3. NASA Fact Sheets ... 48
19. Selbstständigkeitserklärung ... 49
20. Kurzzusammenfassung ... 50

1. Einleitung

Per Zufall habe ich vor einem Jahr in YouTube eine Gestirns Blume gesehen, welche auf den Planetenbewegungen Erde-Venus beruhen soll. Dieses Muster hat eine faszinierende Ausstrahlung, welche mich nicht mehr losgelassen hat und von Zeit zu Zeit wieder in meiner Erinnerung auftauchte. Im Zuge der ersten Nachforschungen, stiess ich auf Johannes Keplers „Harmonice Mundi", welche er 1619 fertigstellte und als Krönung seiner Forschungen ansah. In der „Harmonice Mundi" vertritt er die Meinung, dass Gott die Welt nach harmonischen Prinzipien erschaffen hat. Bereits 1596 zeigt Kepler seine erste Vorstellung der Himmelskörpergeometrie im „Mysterium Cosmographicum" auf.

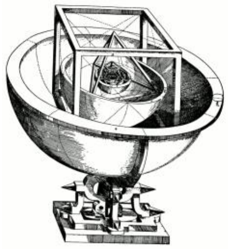


Abbildung 1 - Universum Modell von Kepler

Das Konzept der Maturaarbeit befasste sich mit der Erde-Venus Gestirns Blume, welche ich bereits auf Basis von Planeten- und Gestirns Daten rekonstruieren konnte, um sicherzustellen, dass es sich nicht um eine Falschinformation aus dem Internet handelt. Auf Basis dieses ersten vielversprechenden Ergebnisses wird nun mit dieser Maturaarbeit ein Computerprogramm erstellt, welches alle Planeten unseres Sonnensystems zueinander in Beziehung stellen kann.

Ziel dieser Arbeit soll es nun sein kritisch und rein wissenschaftlich aufzuzeigen, ob es sich bei der Erde-Venus-Blume um einen Zufall oder evtl. doch um eine höhere Ordnung handelt. Dies wird mit Hilfe von weiteren Planetenpaarkombinationen graphisch ausgewertet, indem die Verbindungslinie der beiden Planetenmittelpunkte zeitlich getaktet graphisch dargestellt werden. Handelt es sich um eine höhere Ordnung, wie es Johannes Kepler vermutet, so müssten ebenfalls harmonische Muster entstehen und eine Beziehung hergeleitet werden können.

2. Johannes Keplers Geschichte

Der Astronom Johannes Kepler zählt zu den Vätern der modernen Naturwissenschaft. Er entdeckte als erster die Gesetze der Planetenbewegungen und ermöglichte die Erforschung der Gestirne und des Universums im Allgemeinen, so wie wir es heute kennen. Ebenso legte Johannes Kepler die Basis mit seinem dritten Keplerschen Gesetz für die Gravitationsgesetze von Newton. Johannes Kepler hat sich für das heliozentrische Weltbild entschieden, da es die Schleifenbewegungen der Planeten erklärt und sich die Sonne im Zentrum des Weltalls befindet.

Abbildung 2 - Portrait von Johannes Kepler im Alter von 39 Jahren

Johannes Kepler wurde am 27. Dezember 1571 in Weil der Stadt geboren. 1577 tritt Johannes Kepler in die Lateinschule von Leonberg. Im Jahr 1589 tritt er das Theologiestudium am Evangelischen Stift in Tübingen an, welches er 1591 mit der Beförderung zum Magister (Akademischer Grad) abschloss.

Von 1594 bis 1600 arbeitete Johannes Kepler als Landschaftsmathematiker und Lehrer an der Evangelischen Stiftsschule in Graz. Hier in Österreich zieht Johannes Kepler der Grundgedanke der Weltharmonie (er nennt es Weltgeheimnis) in Betracht. Er geht von dem Grundgedanken aus, dass alles im Universum in einer vollkommenen Harmonie zueinandersteht. Am 30. September 1600 wird Johannes Kepler aus Konfessionsgründen aus Graz ausgewiesen.

Daraufhin reist er nach Prag und nimmt eine Stellung als Assistent von Tycho Brahe an. Die Zusammenarbeit der beiden Wissenschaftler in Prag war aber schwierig, da ihre Fähigkeiten verschieden waren und Tycho Brahe nicht ganz die Ansichten von Johannes Kepler teilte. 1601 stirbt Tycho Brahe unvorhersehbar worauf Johannes Kepler zum kaiserlicher Hofmathematiker befördert wird. 1604 präsentiert Kepler dem Kaiser sein Werk ‚Optischer Teil der Astronomie'. In diesem Werk wird die Theorie der Optik erläutert, so dass man ab diesem Zeitpunkt Fernrohre berechnen konnte und diese nicht durch Ausprobieren herstellen musste. In Prag vertiefte Johannes Kepler seine astronomischen Erkenntnisse (Beobachtung einer Supernova, partielle Sonnenfinsternis, Halley'scher Komet, Jupitermonde, etc.).

Nach dem Tod 1611 seiner Frau und der Entmachtung von Kaiser Rudolph II übersiedelt Johannes Kepler 1612 nach Linz und tritt dort eine Stelle als Mathematiker an. 1613 heiratet Johannes Kepler seine zweite Frau Susanna Reuttinger. Hier in Linz hat Kepler viel Zeit sich der Astronomie zu widmen, wobei er zwischendurch auch ganz praktische Lösungen entwickelte, wie die der Fassgeometrie, eine Formel welche das genaue Volumen eines Fasses bestimmt und für den Handel benötigt wurde, da immer wieder unterschiedliche Fässer verwendet wurden und das genaue Volumen nie ganz klar war. Die Zeit in Linz war immer wieder mit Schwierigkeiten verbunden, so musste Kepler immer wieder hinnehmen, dass sein Gehalt im Rückstand war. Zeitweise wurde sogar seine Bibliothek beschlagnahmt. 1618 findet Johannes Kepler das dritte Keplersche Gesetz und 1619 veröffentlicht er seine Theorie der Weltharmonik (Harmonices Mundi). 1624 schliesst Kepler die Arbeiten an dem Manuskript der Rudolphinischen Tafeln (Ephemeriden Tabellen) ab.

Der Oberösterreichische Bauernkrieg veranlasste Johannes Kepler im Jahr 1626 Linz zu verlassen und nach Ulm umzusiedeln. 1627 werden seine Rudolphinischen Tafeln veröffentlicht und auf der Buchmesse in Frankfurt am Main vorgestellt. Nach vielen beschwerlichen Reisen stirbt Johannes Kepler am 15. November 1630 in Regensburg. Auch nach seinem Tod werden von seinem Sohn noch nicht veröffentlichte Werke publiziert (z.B. 1634 die Mondastronomie).

3. Unser Sonnensystem

Unser Sonnensystem besteht aus der Sonne, ihren acht Planeten und deren Monden, den Zwergplaneten und Millionen von Kleinkörpern wie beispielsweise Asteroiden und Kometen. Sie alle kreisen um die Sonne. Die Sonne ist der Zentralstern unseres Sonnensystems und macht ca. 99% der Gesamtmasse des Systems aus. Die der Sonne am nächsten stehenden, terrestrischen Planeten sind Merkur, Venus, Erde und Mars. Weiter draussen befinden sich die Gasplaneten Jupiter, Saturn, Uranus und Neptun (früher auch noch Pluto[1]). Zwischen Mars und Jupiter befindet sich der Asteroidengürtel, die wichtigsten Asteroiden sind Vesta, Juno, Ceres und Pallas.

[1] Pluto wurde früher auch als Planet angesehen, ist aber durch eine Änderung der Planetendefinition im Jahr 2006 heute kein Planet im Sinne der Definition.

Da die Daten wie; Planetenumlaufzeit, Planetendurchmesser, Entfernung zur Sonne sowie die Ellipsengeometrie der Planeten ziemlich genau bekannt sind, beschränken wir uns auf diese 8 Planeten, bei unserer Überprüfung.

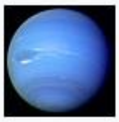
Erdähnliche Planeten				Gasplaneten			
Merkur	Venus	Erde	Mars	Jupiter	Saturn	Uranus	Neptun
							

Abbildung 3 - Planeten des Sonnensystems - nicht massstabstreue Darstellung der Planeten

4. Gedanklicher Abriss der Vorgehensweise

Je tiefer ich mich in die Materie hineingearbeitet habe, desto mehr Unklarheiten haben sich ergeben. Mehr und mehr unbekannte Ausdrücke sind aufgetaucht, z.B. siderisches Jahr, numerische Exzentrizität, lineare Exzentrizität, AE, Perihel, Aphel, usw.

Dieses Kapitel soll die Gedankengänge widergeben, worauf die nachfolgenden Kapitel basieren und die Basis der errechneten Graphiken bilden. Dieses Kapitel erhebt nicht den Anspruch den wissenschaftlichen Gepflogenheiten eines rein naturwissenschaftlichen Manuskriptes zu entsprechen, sondern die Ordnung meiner Gedankengänge widerzuspiegeln. Alle nachfolgenden Kapitel wurden unter Beachtung der Regeln eines naturwissenschaftlichen Manuskriptes erstellt und sollten diesen auch gerecht werden.

Begeben wir uns nun in die Unendlichkeit der Gedankengänge. Wir stellen uns also einen Planeten vor, z.B. die Venus, welche sich auf einer elliptischen Bahn um die Sonne bewegt. Die Bahngeschwindigkeit des Planeten ist aber nicht gleichmässig (siehe 2. Keplersches Gesetz), die elliptische Bahngeometrie (kleine und grosse Ellipsenhalbachse, Achsenneigung, etc.) ist bekannt. Im Prinzip wissen wir nun alles Notwendige, um mit den Berechnungen der beiden Bahnen zu beginnen und die Verbindungslinie zwischen den Planetenpositionszentren zu jedem Zeitpunkt berechnen und zeichnen zu können.

Nun tun sich aber noch weitere Probleme auf. Wenn eine zweidimensionale Darstellung gewählt wird, wohin ist dann die Projektionsebene zu legen? Zwischen die beiden Planetenbahnebenen oder soll als Referenz die Ellipsenbahnebene der Erde definiert werden? Oder macht es mehr Sinn eine dreidimensionale Darstellung anzuwenden? Das Resultat (die Graphik) sollte hochauflösend vorliegen und auch Ausdrucke über einen Plotter oder Grossfarbdrucker in vernünftiger Qualität ermöglichen.

Um die Datenberechnung sinnvoll herzuleiten, mussten umfangreiche Prüfungen und damit auch Berechnungen über die Relevanz der einzelnen Parameter durchgeführt werden, um das Programm auf ein Niveau zu bringen, welches einer Maturaarbeit entspricht und nicht in einer Doktorarbeit endet. Die Vereinfachungen müssen transparent und nachvollziehbar sein. Die folgenden Punkte mussten erst herausgearbeitet bzw. abgeklärt werden;

- Umlaufgeschwindigkeit in Abhängigkeit der Bahnposition zur Sonne
- Neigungswinkel der Planetenbahnen zueinander
- Ausgangsposition der beiden Planeten (Datum, Planeten in Konjunktion, etc.)
- Zwei- oder dreidimensionale Darstellung der Verbindungsgeraden der Planetenzentren
- Prüfung ob die elliptischen Bahnen nicht als Kreis angenommen werden können
- Tagesschritte festlegen für die Positionsberechnung, damit ein klares Muster entsteht
- usw.

5. Die Keplerschen Gesetze

5.1. Das 1. Keplersche Gesetz

Jeder Planet bewegt sich auf einer elliptischen Bahn. In den Brennpunkten dieser elliptischen Bahn steht die Sonne.

$$\boldsymbol{Brennpunkte\ F_1\ \ oder\ \ F_2}$$ (siehe Abbildung 4)

5.2. Das 2. Keplersche Gesetz

Die Vektorfläche Sonne-Planet überstreicht in gleichen Zeiteinheiten dieselben Flächen, d.h. dass ein Planet in Sonnennähe (z.B. im Perihel) sich schneller bewegt und somit pro Zeiteinheit einen längeren Weg zurücklegt, als wenn er sich in der Sonne entfernteren Bereichen befindet (z.B. im Aphel). Die während einer Zeiteinheit überstrichene Fläche (Sonne-Planet) die durch den Anfangspunkt und den Endpunkt der Messung entstehen, sind pro Zeiteinheit immer identisch, unabhängig von der Position des Planeten auf der Bahn.

$$Fläche\ \boldsymbol{A_1} = Fläche\ \boldsymbol{A_2}$$ (siehe Abbildung 4)

5.3. Das 3. Keplersche Gesetz

Johannes Kepler veröffentlichte 1618 das dritte der nach ihm genannten Gesetze auf Basis von Beobachtungen der Umlaufdaten des Mars. Im Jahr 1619 veröffentlicht Johannes Kepler das berühmte Werk *‚Harmonices mundi'* in dem er erstmals die Gesetzmässigkeit publizierte.

Das Gesetz lautet wie folgt;

Die Quadrate der Planetenumlaufzeiten verhalten sich wie die dritte Potenz der jeweiligen grossen Halbachsen a der entsprechenden Ellipsenbahnen.

$$\frac{(\boldsymbol{T_1})^2}{(\boldsymbol{T_2})^2} = \frac{(\boldsymbol{a_1})^3}{(\boldsymbol{a_2})^3}$$

Johannes Kepler hatte aufgrund seiner Beobachtungen festgestellt, dass für alle Planeten der Wert C (Kepler Konstante) identisch ist.

$$\frac{T^2}{a^3} = C\ (Kepler\ Konstante)$$

In der nachfolgenden Tabelle werden die mit den heutigen Werten berechneten Keplerkonstanten für einen Teil der Planeten in unserem Sonnensystem ausgerechnet. Die Konstante wird in Beziehung mit verschiedenen Planeten ausgerechnet, um sehen zu können, ob diese immer den gleichen Wert ergibt.

Tabelle 1; Planetendaten zum 3.Keplerschen Gesetz

Planet bzw. Gestirn	$\frac{T^2}{a^3}$
Merkur	1.002
Venus	1.000
Erde	1
Mars	0.999
Jupiter	0.999
Saturn	0.995

T = siderische Umlaufzeit des Planeten in Erdenjahren
a = Länge der grossen Halbachse der elliptischen Umlaufbahn in astronomischen Einheiten (AE)
AE = Abstand zwischen Sonne und Erde (= 1 AE)

Wie man sieht gibt es nur geringe Abweichungen.

Nachfolgend, eine Graphik zum besseren Verständnis der Keplerschen Gesetze.

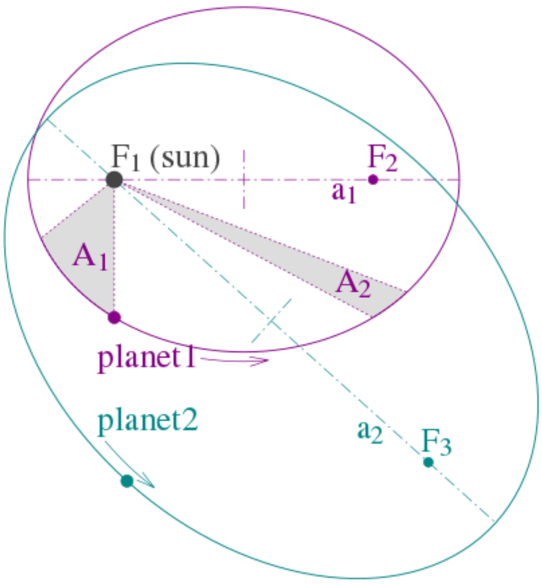


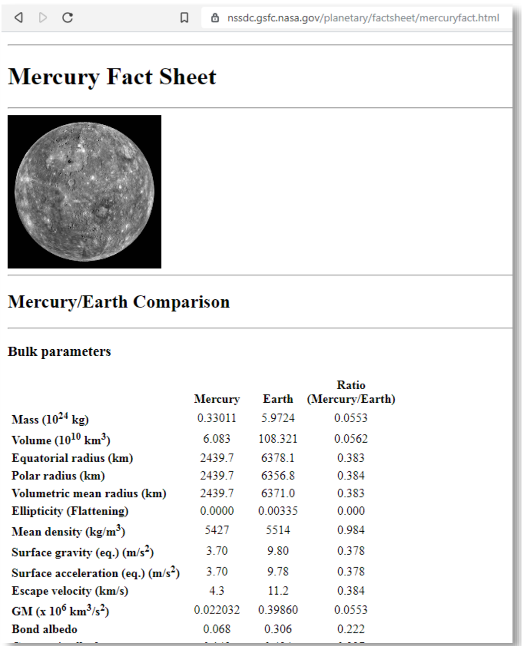
Abbildung 4 - Darstellung der Keplerschen Gesetze

6. Berechnung der Planetenbahnen

Im Internet konnten man verschiedene Quellen finden (Wikipedia, Sternwarten, Universitäten, etc.) welche teilweise gegensätzliche Angaben enthielten, so dass nicht auf eindeutige Planetendaten zurückgegriffen werden konnte. Bei den Angaben zur Erde wurde das Erdenjahr von 365 Tagen bis 365.4 Tagen definiert, so dass sehr schnell klar wurde, dass diese Datenquellen für die Berechnungen ungeeignet sind. Im Zuge der Recherche konnten zwei vertrauenswürdige Quellen ermittelt werden, erstens die eines französischen Institutes VSOP (Variations Séculaires des Orbites Planétaires) und zweitens die Daten der amerikanischen NASA (National Aeronautics and Space Administration).

Das VSOP-Modell wurde von den Wissenschaftlern des ‚Bureau des Longitudes' in Paris (Frankreich) entwickelt und wird von diesen auch weitergeführt und ständig verbessert. Die neusten Messungen werden kontinuierlich integriert, um ein möglichst genaues Ergebnis zu erhalten. Die VSOP2013-Dateien enthalten die Reihen der elliptischen Elemente für die 8 Planeten Merkur, Venus, Erde-Mond-Schwerpunkt, Mars, Jupiter, Saturn, Uranus, Neptun und für den Zwergplaneten Pluto.

Die NASA ist eine unabhängige Behörde der Bundesregierung der Vereinigten Staaten, die für das zivile Raumfahrtprogramm sowie für die Luft- und Raumfahrtforschung zuständig ist. Beide nationalen Institute/Behörden liefern frei zugänglich Daten der Planeten und Gestirne. Da es sich bei den französischen Daten um Rohdaten handelt welche mit geeigneten Computerprogrammen erst in eine verwendbare Form gebracht werden müssen, habe ich mich entschieden die Daten der NASA zu verwenden, welche leicht verständlich auf dem Internet verfügbar sind; (https://ssd.jpl.nasa.gov/horizons.cgi?s_target=1#top [Stand: 17.11.2019] und https://nssdc.gsfc.nasa.gov/planetary/factsheet/ [Stand: 17.11.2019]).


nssdc.gsfc.nasa.gov/planetary/factsheet/mercuryfact.html

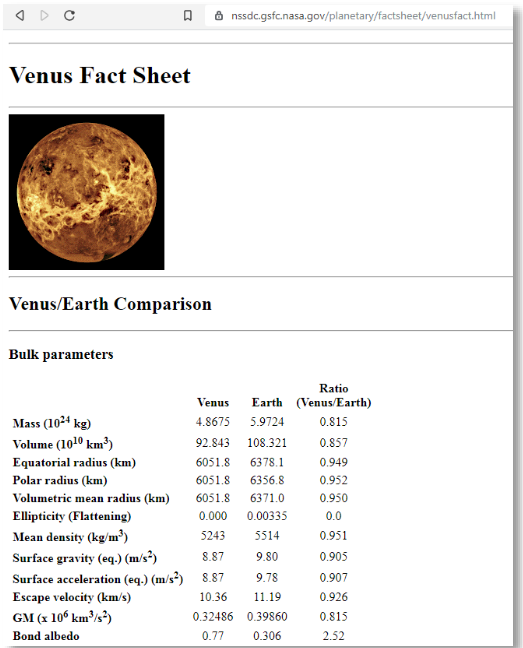
Mercury Fact Sheet

Mercury/Earth Comparison

Bulk parameters

	Mercury	Earth	Ratio (Mercury/Earth)
Mass (10^{24} kg)	0.33011	5.9724	0.0553
Volume (10^{10} km^3)	6.083	108.321	0.0562
Equatorial radius (km)	2439.7	6378.1	0.383
Polar radius (km)	2439.7	6356.8	0.384
Volumetric mean radius (km)	2439.7	6371.0	0.383
Ellipticity (Flattening)	0.0000	0.00335	0.000
Mean density (kg/m^3)	5427	5514	0.984
Surface gravity (eq.) (m/s^2)	3.70	9.80	0.378
Surface acceleration (eq.) (m/s^2)	3.70	9.78	0.378
Escape velocity (km/s)	4.3	11.2	0.384
GM (x 10^6 km^3/s^2)	0.022032	0.39860	0.0553
Bond albedo	0.068	0.306	0.222

Abbildung 5 - Merkur Fact Sheet


nssdc.gsfc.nasa.gov/planetary/factsheet/venusfact.html

Venus Fact Sheet

Venus/Earth Comparison

Bulk parameters

	Venus	Earth	Ratio (Venus/Earth)
Mass (10^{24} kg)	4.8675	5.9724	0.815
Volume (10^{10} km^3)	92.843	108.321	0.857
Equatorial radius (km)	6051.8	6378.1	0.949
Polar radius (km)	6051.8	6356.8	0.952
Volumetric mean radius (km)	6051.8	6371.0	0.950
Ellipticity (Flattening)	0.000	0.00335	0.0
Mean density (kg/m^3)	5243	5514	0.951
Surface gravity (eq.) (m/s^2)	8.87	9.80	0.905
Surface acceleration (eq.) (m/s^2)	8.87	9.78	0.907
Escape velocity (km/s)	10.36	11.19	0.926
GM (x 10^6 km^3/s^2)	0.32486	0.39860	0.815
Bond albedo	0.77	0.306	2.52

Abbildung 6 - Venus Fact Sheet

Die nachfolgenden zwei Beispiele bilden einen Teil der von der NASA publizierten Planetendaten für den Planeten Merkur und Venus gemäss ‚Fact Sheet' [2] ab;

Alle verwendeten Fact Sheets der NASA sind im Anhang 8.3. referenziert.

Unter der Rubrik , *Orbital parameters* ' kann man alle Planeten-Daten wie Umlaufzeit, numerische Exzentrizität, Perihel, Aphel, grosse Ellipsenhalbachse, etc. finden. Die nachfolgende Tabelle beruht auf diesen NASA-Daten und stellt das relevante Extrakt daraus dar, welches für die weiteren Berechnungen verwendet wurde.

Zusammenstellung der Planetendaten;

Tabelle 2; Planetendaten der NASA sowie eigene Berechnungen grün hinterlegt

Planet bzw. Gestirn	Umlaufzeit in Tagen	Numerische Exzentrizität der Umlaufbahn	Perihel in [x10⁶ km]	Aphel in [x10⁶ km]	Grosser Ellipsenradius a[3] in [x10⁶ km]	Kleiner Ellipsenradius b[4] in [x10⁶ km]
Merkur	87.969	0.2056	46.00	69.82	57.91	56.67
Venus	224.701	0.0067	107.48	108.94	108.21	108.21
Erde	365.256[5]	0.0167	147.09	152.10	149.60	149.58
Mars	686.980	0.0935	206.62	249.23	227.92	226.92
Jupiter	4'332.589	0.0489	740.52	816.62	778.57	777.64
Saturn	10'759.22	0.0565	1'352.55	1'514.50	1'433.53	1'431.24
Uranus	30'685.4	0.0457	2'741.30	3'003.62	2'872.46	2869.46
Neptun	60'189.0	0.0113	4'444.45	4'545.67	4'495.06	4494.77

Da nirgends auf den Planetendatenblättern der Ellipsenparameter b gelistet war, musste dieser Wert aus den Angaben errechnet werden. Obwohl der Ellipsenparameter a auf den Planetendatenblättern der NASA gelistet war, wurde dieser vollständigkeitshalber ausgerechnet und danach mit den Daten der NASA verglichen. Der Ellipsenparameter a hatte den gleichen Wert, wie der des ausgerechneten Werts. Die Exzentrizität entsteht, indem die Umlaufbahnen der Planeten sich im Rhythmus von vielen Jahrtausenden, so im Fall der Erde, etwa 100 000 Jahren, durch die Wirkung benachbarter Planeten von einem nahezu perfekten Kreis[6] zu einer Ellipse verformt wird. Zurzeit beträgt die numerische Exzentrizität der Erde 0,0167[7] und ist sehr gering– mit dem blossen Auge könnten Sie nicht erkennen, ob es sich um einen Kreis oder eine Ellipse handelt!

Den Ellipsenparameter a kann man anhand des Perihels und dem Aphel berechnen, indem man diese zwei Werte addiert und durch zwei teilt. Zum Beispiel bei der Venus: Perihel mit Aphel zusammengerechnet ergeben 216.42 [x10^6 km]. Wenn dieser Wert nun halbiert wird

[2] https://nssdc.gsfc.nasa.gov/planetary/factsheet/index.html [Stand: 17.11.2019]
[3] Englisch: Ellipse semi-major axe
[4] Englisch: Ellipse semi-minor axe
[5] Die Umlaufzeit nennt man auch siderische Periode und wird im Fall der Erde als 1 siderisches Jahr bezeichnet. Auf Basis dieser Festlegung können nun alle anderen Planeten berechnet werden; z.B. Venus 0.615 siderische Jahre, Jupiter 11.862 siderische Jahre, Merkur 0.241 siderische Jahre, etc.
[6] Der Kreis ist ein Sonderfall der Ellipse, hier vereinen sich die beiden Brennpunkte zu einem Brennpunkt und damit ist die Exzentrizität = 0
[7] Der Exzentrizitätswert der Erde wurde aus der Tabelle oben entnommen. Hier können Sie auch die entsprechenden Werte der anderen Planeten vergleichen.

erhält man 108.21 [x10^6 km], was dem Ellipsenparameter a entspricht. Obwohl man den Ellipsenparameter a so berechnen könnte, wurden die Daten der NASA verwendet, da sie diese zur Verfügung stellen.

Das nachfolgende Bild zeigt eine typische Ellipse mit Ihren charakteristischen Parametern.

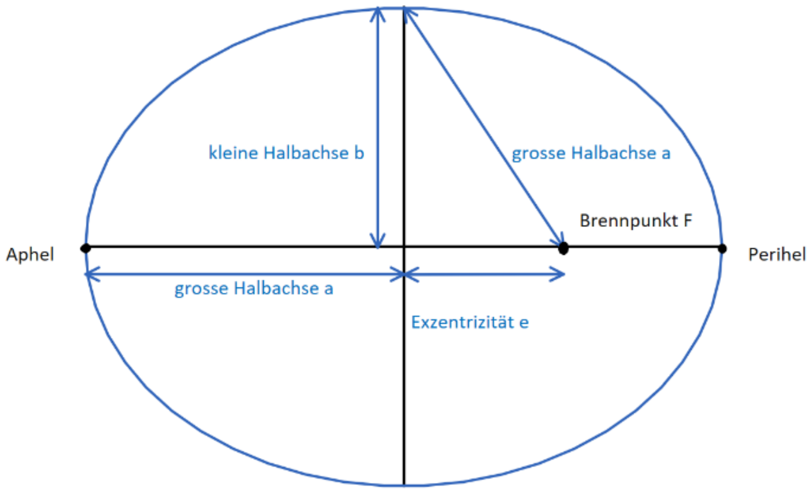


Abbildung 7 - Ellipsengeometrie Skizze

Gemäss der Ellipsenmathematik kann der kleine Ellipsenradius aus dem Wert a und der numerischen Exzentrizität e der Ellipse berechnet werden. Die nachfolgende Formel gibt die Beziehung zur Berechnung von b (=kleiner Halbellipsenradius) wieder;

$$b = \mathrm{a}\sqrt{1-\mathrm{e}^2}$$

Die nachfolgenden beiden Formeln ergeben die numerische Exzentrizität e einer Ellipse;

$$\varepsilon = \frac{\sqrt{a^2-\mathrm{b}^2}}{a}$$

oder umgeformt;

$$\varepsilon = \sqrt{1-\left(\frac{b}{a}\right)^2}$$

Neben der numerischen Exzentrizität gibt es auch noch die lineare Exzentrizität e, welche sich aus der Formel nach Pythagoras berechnen lässt;

$$e = \sqrt{a^2-b^2}$$

als Verhältnis der linearen Exzentrizität geteilt durch die grosse Halbachse a definiert die numerische Exzentrizität und ist somit eine dimensionslose Zahl;

$$\frac{e}{a} = \varepsilon$$

Mit Hilfe dieser genannten Ellipsenformeln und den Angaben der NASA, wurden alle fehlenden Daten ermittelt. Die berechneten Daten wurden in der Tabelle oben hellgrün hinterlegt.

Die x,y-Koordinaten führen zu einer relativen Position auf einer Ebene und nicht als absolute Position im Raum. Die absolute Position im Raum kann berechnet werden, wenn die Planetenellipsenebene in Beziehung zur Sonnen-Erde-Ebene gesetzt wird. Die orbitalen Neigungswinkel der Planetenbahnen wurden ebenfalls aus den Angaben von NASA entnommen und in der Tabelle unten gelistet.

Tabelle 3; Orbitale Bahnneigungswinkel

Planet bzw. Gestirn	Orbitale Bahnneigung in Grad
Merkur	7.004
Venus	3.394
Erde	0.000
Mars	1.850
Jupiter	1.308
Saturn	2.488
Uranus	0.774
Neptun	1.774

Da es sich bei den Neigungswinkeln um geringe Gradzahlen handelt und die Projektion auf eine Ebene sich nur geringfügig verformt, wurde die dreidimensionale Lage der Planeten im Raum vernachlässigt.

Als weiterer Punkt gilt es die Bahngeschwindigkeiten während der Planetenbewegung auf der Ellipse, welche im Aphel und Perihel unterschiedlich ist, zu analysieren. Die nachfolgende Tabelle zeigt die maximale und minimale Geschwindigkeit der Gestirne, bezogen auf den Perihel und den Aphel. Diese beiden Positionen geben die langsamste und schnellste Geschwindigkeit wider und werden für die Beurteilung der Relevanz herangezogen. Im Perihel hat der Planet die maximale Geschwindigkeit und im Aphel die minimale. Die berechneten Daten wurden in Tabelle 4 ebenfalls hellgrün hinterlegt.

Die berechnete Planetengeschwindigkeitsabweichung gemäss Tabelle 4 zeigt, dass die Geschwindigkeitsdifferenzen 1 bis 10% von der mittleren Bahngeschwindigkeit abweichen, ausser im Fall vom Merkur und Mars, da sind die Unterschiede deutlich grösser. Beim Merkur sind die Geschwindigkeitsunterschiede mit über 42% markant.

Die Berücksichtigung der Bahngeschwindigkeit zur Berechnung der Planetenbahnbewegungen wurde als nicht notwendig erachtet und so dass die von der NASA angegebene mittlere Planetengeschwindigkeit für die weiteren Berechnungen Verwendung fand.

Somit wird das 2. Keplersches Gesetz in der Programmierung nicht berücksichtigt.

Da die Abweichung der Merkurbahngeschwindigkeit mit 42.48% gegenüber der mittleren Planetengeschwindigkeit recht hoch ist, muss hier mit verzerrten Bildern gerechnet, so dass diese Figuren für die Auswertung mit Vorsicht zu verwenden sind.

Tabelle 4; Geschwindigkeitsverlauf ausgesuchter Planeten auf der Ellipsenbahn

Planet bzw. Gestirn	Durchschnittliche Geschwindigkeit [km/s]	Maximale Geschwindigkeit [km/s]	Minimale Geschwindigkeit [km/s]	Differenz der Max. und Min. Geschwindigkeit	Abweichung [%]
Merkur	47.36	58.98	38.86	20.12	42.48
Venus	35.02	35.26	34.79	0.47	1.34
Erde	29.78	30.29	29.29	1	3.36
Mars	24.07	26.50	21.97	4.53	18.82
Jupiter	13.06	13.72	12.44	1.28	9.80
Saturn	9.68	10.18	9.09	1.09	11.26
Uranus	6.80	7.11	6.49	0.62	9.12
Neptun	5.43	5.50	5.37	0.13	2.40

In diesem Absatz gehen wir auf die Berechnung der Planetenbahnen bezogen auf das Programm ein bzw. die Verbindungslinie der Gestirns Mittelpunkte zweier Planeten zueinander. Dazu müssen wir die Position des Planeten auf seiner elliptischen Bahn berechnen. Hierzu ist aber erst ein Blick auf die Ellipsengeometrie zu werfen. In der nachfolgenden Graphik ist die Planetenposition auf der Ellipsenbahn mit einem roten Punkt eingezeichnet. Die Position ergibt sich nun durch die beiden Winkelfunktionen Sinus bzw. Cosinus und die beiden Ellipsenhalbachsen a und b in Abhängigkeit des Winkels bzw. der Umlaufzeit in Tagen, aus derer die Winkelschritte berechnet werden können. Die Position des Planeten lässt sich demnach nach den beiden folgenden Funktionen als x- und y-Wert berechnen und somit der Bewegungsverlauf graphisch darstellen;

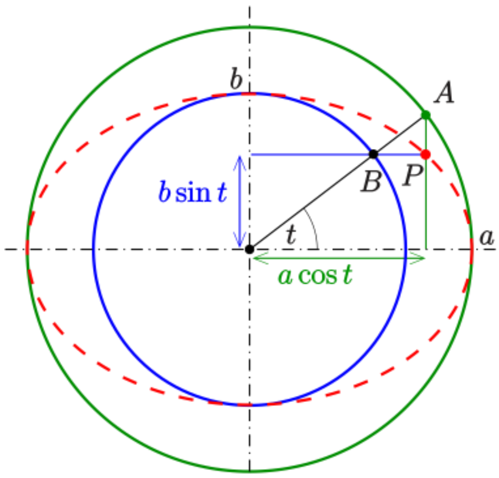


Abbildung 8 - Darstellung der Ellipsengeometrie

$$x = a * cos\ (t)$$

$$y = b * \sin\ (t)$$

Der Winkel t lässt sich über die Planetenumlaufzeit in Tagen, welche einem Winkel von 360° entspricht, für jeden Tag genau berechnen. Die Angaben für die Bahnumlaufzeit wurden ebenfalls aus den Angaben der NASA entnommen und für die Berechnungen verwendet.

7. Planetenbahnen in Beziehung gesetzt

Gemäss dem vorherigen Kapitel kann man nun die Planetenmittelpunktpositionen für jeden Tag als x,y-Koordinaten berechnet. Berechnet man nun zum selben Zeitpunkt die x,y-Koordinaten eines weiteren Planeten so kann man die beiden Planetenmittelpunktpositionen durch eine Linie verbinden. Diese Verbindungslinie wird nun in regelmässigen Abständen, z.B. alle 3 Tage, berechnet und in ein Koordinatensystem eingezeichnet. Über einen Zeitraum von mehreren Jahren entsteht nun eine Zeichnung, die zufällig und ohne Ausbildung eines Musters aussehen wird, sollte keine höhere Ordnung existieren. Eine erste Auswertung wurde bereits im Konzept ansatzweise graphisch erstellt. Das nachfolgende Bild zeigt die ersten Berechnungen der Erde-Venus Verbindungslinie über einen Zeitraum von 180 Tagen in 3-Tagesschritten.

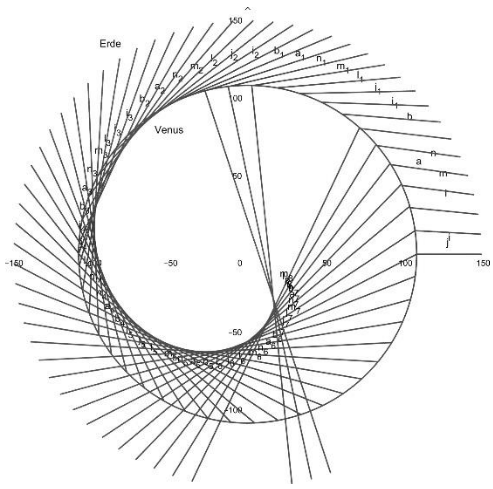


Abbildung 9 - Graphik der Erde-Venus Verbindungslinien über ein halbes Jahr

Aus dem Bild kann man erkennen, dass sich eine harmonische Kontur auszubilden scheint. Um nun einen längeren Zeitraum abbilden zu können, muss ein geeignetes Computerprogramm entwickelt werden, da es sonst nicht möglich ist, einen tieferen Einblick auf den Konturenverlauf zu erhalten. Ohne diesen längerfristigen Blick auf den Konturenverlauf kann keine Beziehung nachgewiesen werden, wie es Johannes Kepler annimmt und mit einer Theorie zu beweisen versuchte. Das Computerprogramm basiert auf der Ellipsenlaufbahn der beiden Planeten und die tagesgenaue Bestimmung der beiden Planetenkoordinaten auf dieser Planetenbahn des Gestirns. Die x,y-Koordinaten wurden bereits im vorherigen Kapitel bestimmt und die Bewegungsgeschwindigkeit ergibt sich durch die Ellipsenbahnumlaufzeit um eine vollständige Kreisbewegung zu absolvieren. Diese Daten sind im entsprechenden Datenblatt (Englisch; Fact Sheet) aufgelistet und müssen nur noch in einer Funktion kombiniert werden, welche die Basis des Simulationsprogramms bildet.

Die nachfolgenden zentralen Befehlszeilen bilden diese Funktion zusammenfassend ab;

xLineP1 = (math.cos(x1))*a1

yLineP1 = (math.sin(x1))*b1

xLineP2 = (math.cos(x2))*a2

yLineP2 = (math.sin(x2))*b2

line(xLineP1, yLineP1, xLineP2, yLineP2)[8]

Legende zur Befehlszeile;
a1…grosse Halbachse von Planet 1
b1…kleine Halbachse von Planet 1
a2…grosse Halbachse von Planet 2
b2…kleine Halbachse von Planet 2
x1…Winkel des Planeten 1 vom Ausgangsort
x2…Winkel des Planeten 2 vom Ausgangsort

Es erklärt sich von selbst, dass die oben genannte Funktion je nach Programmiersprachen, Winkeleingaben ins Bogen- bzw. Gradmass umgerechnet werden müssen, damit ein auswertbares Ergebnis resultiert. Es ist auch klar, dass mit dieser Funktion allein noch keine brauchbaren Graphiken erzielt werden können, aber sie stellt das Herz des Programms dar. Ausserdem kann auch der Mittelpunkt dieser Line als Einzelpunkt dargestellt werden.

Als weiterer Punkt sind die Tagesschritte anzusehen. Eine Funktion konnte nicht hergeleitet werden, um die Tagesschritte und die Zeitperiode für ein harmonisches Muster zu berechnen. Die unterschiedlichen Entfernungen zueinander und je nach Ellipsenbahn auch die grossen Unterschiede in der Umlaufgeschwindigkeit macht es schwer eine Funktion abzuleiten. Im Fall der Planetenkombination Venus-Erde konnte eine Beziehung auf Basis der Bahnumlaufzeiten (365 und 224 Tage) und aus dem daraus resultierenden Verhältnis 0.6 (=224 Tage/365 Tage) durch Verdopplung bis eine gerade Zahl entsteht ($0.6*\mathbf{2*2*2} \cong 5$), ausfindig gemacht werden. Bildet man nun den Multiplikationswert **2∗2∗2**=8 so ergibt sich die Anzahl Umlaufjahre welche gerechnet werden müssen, um eine vollständige Figur (in dem Fall eine Blume, siehe Abbildung 32) zu erhalten. Die Erde hat eine Umlaufzeit von 365.256 Tagen und die Venus eine Umlaufzeit von 224.701 Tagen. Das ergibt ein Verhältnis von ungefähr 13:8. Die Differenz beträgt also 5. Wenn diese nun mit der synodischen[9] Umlaufzeit der Venus multipliziert wird, so erhält man als Resultat ungefähr acht Jahre.

Leider konnte diese Gesetzmässigkeit nicht auf die anderen Planetenkombinationen angewendet werden, so dass die anfänglichen Überlegungen und Vermutungen nicht zutreffend sind. Aus Zeitgründen konnte nicht weiter auf diesen Aspekt eingegangen werden, was dazu führte, dass empirisch die Tagesschrittweite und Zeitdauer ermittelt werden mussten. Dies erfolgte mit dem Programm, womit die ideale Schrittlänge und Schrittanzahl für jede Planetenkombination experimentell ermittelt wurde. Diese Parameter sind in der folgenden Tabelle 5 aufgelistet. Die obere Zahl im Tabellenfeld definiert die Schrittweite in Tagen und die Zahl darunter die Anzahl Wiederholungen (Zyklen), die dabei gemacht wurden. Die Gesamtzeit in Tagen kann errechnet werden, indem man diese zwei Zahlen multipliziert. Zusätzlich wurde je

[8] Basierend auf dem Befehlscode von TigerJython Version 2.15.08
[9] Zeitraum, welcher der Planet benötigt, um von einer Konjunktionsposition zur nächsten zu gelangen

nach Gesamtdauer entschieden, ob die Graphik durch Raumgeraden (RG [10]) oder durch Geradenmittelpunkte (SP [11]) dargestellt werden soll, um eine Figur zu erkennen (dritte Zeile).

Tabelle 5 ; Matrix der empirisch ermittelten Planetenkombinationseinstellungen

Planeten-kombination	Venus	Erde	Mars	Jupiter	Saturn	Uranus	Neptun
Merkur	3 Tage 680 Zyklen RG & SP	3 Tage 850 Zyklen RG & SP	3 Tage 1150 Zyklen RG & SP	5 Tage 867 Zyklen RG & SP	10 Tage 1075 Zyklen SP	30 Tage 1030 Zyklen SP	50 Tage 1205 Zyklen SP
Venus		3 Tage 974 Zyklen RG & SP	5 Tage 2350 Zyklen RG & SP	10 Tage 435 Zyklen RG & SP	10 Tage 1080 Zyklen SP	15 Tage 2047 Zyklen SP	50 Tage 1205 Zyklen SP
Erde			5 Tage 1120 Zyklen RG & SP	10 Tage 438 Zyklen RG & SP	10 Tage 1095 Zyklen SP	20 Tage 1535 Zyklen SP	50 Tage 1205 Zyklen SP
Mars				10 Tage 1305 Zyklen RG & SP	10 Tage 1100 Zyklen SP	50 Tage 620 Zyklen SP	50 Tage 1205 Zyklen SP
Jupiter					20 Tage 1085 Zyklen SP	50 Tage 7350 Zyklen SP	100 Tage 607 Zyklen SP
Saturn						50 Tage 4305 Zyklen SP	300 Tage 1000 Zyklen SP
Uranus							550 Tage 2800 Zyklen SP

Bei gewissen Kombinationen wie z.B. Merkur-Uranus bildeten sich anfangs Blumenmuster heraus, diese werden aber bedingt durch die hohe Bahngeschwindigkeit des inneren Planeten (Merkur) im Verhältnis zum äusseren Planeten (Uranus) oftmals wieder überzeichnet und sind somit nicht mehr sichtbar.

Eine bessere Darstellung und als Lösung dieses vorgängig beschriebenen Problems, ist in diesen Fällen das Zeichnen von Mittelpunkten (=Linienschwerpunkte) der ermittelten Verbindungslinie der beiden Planetenpositionen. Diese Punkte gestatten eine Darstellung, ohne dass die vielen Linien ineinanderlaufen und eine Figur so sichtbar macht, da alle unnötigen Informationen extrahiert wurden. Diese optionale Zusatzfunktion soll ebenfalls im Programm berücksichtigt werden, damit in den Fällen von Planetenkombinationen, wie z.B. Merkur-Uranus die erhaltenen Figuren auch sinnvoll beurteilt werden können.

Als letzter wichtiger Punkt ist die Anfangsposition der beiden Planeten zu definieren, bei welcher wir die Graphik anfangen zu zeichnen. Die Gestirnsephemeridendaten, welche ebenfalls auf der NASA Internetseite (https://ssd.jpl.nasa.gov/horizons.cgi#top [Stand: 17.11.2019]) abrufbar sind, können dazu verwendet werden. Das nachfolgende Bild zeigt einen Auszug der Venus-Ephemeriden, welche ausgegeben werden.

[10] Verbindungslinie der beiden Planeten, wurde hier Raumgerade benannt ist in unseren Fall aber nur eine 2D-Projektion.
[11] Entspricht dem Schwerpunkt der Raumgerade bzw. auch dem Mittelpunkt der RG.

Results

```
*******************************************************************************
Ephemeris / WWW_USER Sat Nov  2 05:24:08 2019 Pasadena, USA      / Horizons
*******************************************************************************
Target body name: Venus (299)                     {source: DE431mx}
Center body name: Earth (399)                     {source: DE431mx}
Center-site name: GEOCENTRIC
*******************************************************************************
Start time      : A.D. 2019-Nov-02 00:00:00.0000 UT
Stop  time      : A.D. 2019-Dec-02 00:00:00.0000 UT
Step-size       : 1440 minutes
*******************************************************************************
Target pole/equ : IAU_VENUS                       {East-longitude positive}
Target radii    : 6051.8 x 6051.8 x 6051.8 km     {Equator, meridian, pole}
Center geodetic : 0.00000000,0.00000000,0.0000000 {E-lon(deg),Lat(deg),Alt(km)}
Center cylindric: 0.00000000,0.00000000,0.0000000 {E-lon(deg),Dxy(km),Dz(km)}
Center pole/equ : High-precision EOP model        {East-longitude positive}
Center radii    : 6378.1 x 6378.1 x 6356.8 km     {Equator, meridian, pole}
Target primary  : Sun
Vis. interferer : MOON (R_eq= 1737.400) km        {source: DE431mx}
Rel. light bend : Sun, EARTH                      {source: DE431mx}
Rel. lght bnd GM: 1.3271E+11, 3.9860E+05 km^3/s^2
Atmos refraction: NO (AIRLESS)
RA format       : HMS
Time format     : CAL
EOP file        : eop.191031.p200122
EOP coverage    : DATA-BASED 1962-JAN-20 TO 2019-OCT-31. PREDICTS-> 2020-JAN-21
Units conversion: 1 au= 149597870.700 km, c= 299792.458 km/s, 1 day= 86400.0 s
Table cut-offs 1: Elevation (-90.0deg=NO ),Airmass (>38.000=NO), Daylight (NO )
Table cut-offs 2: Solar elongation (  0.0,180.0=NO ),Local Hour Angle( 0.0=NO )
Table cut-offs 3: RA/DEC angular rate (     0.0=NO )
************************************************************************************************************
 Date__(UT)__HR:MN     R.A._(ICRF/J2000.0)_DEC  APmag  S-brt             delta      deldot    S-O-T /r    S-T-O
************************************************************************************************************
$$SOE
 2019-Nov-02 00:00     15 50 37.79 -20 27 57.9   -3.84   0.97 1.56164819531306  -6.5449646  20.9253 /T  29.1826
 2019-Nov-03 00:00     15 55 47.01 -20 46 06.0   -3.84   0.97 1.55785028808865  -6.6059991  21.1675 /T  29.5241
 2019-Nov-04 00:00     16 00 57.43 -21 03 39.7   -3.84   0.98 1.55401733301098  -6.6663165  21.4093 /T  29.8653
 2019-Nov-05 00:00     16 06 09.00 -21 20 38.3   -3.84   0.98 1.55014974437605  -6.7259180  21.6505 /T  30.2064
 2019-Nov-06 00:00     16 11 21.73 -21 37 01.0   -3.84   0.98 1.54624793343766  -6.7848121  21.8913 /T  30.5473
 2019-Nov-07 00:00     16 16 35.58 -21 52 47.3   -3.84   0.98 1.54231230554361  -6.8430102  22.1316 /T  30.8880
 2019-Nov-08 00:00     16 21 50.52 -22 07 56.3   -3.84   0.99 1.53834325894922  -6.9005253  22.3714 /T  31.2286
 2019-Nov-09 00:00     16 27 06.54 -22 22 27.6   -3.85   0.99 1.53434118387505  -6.9573725  22.6106 /T  31.5691
 2019-Nov-10 00:00     16 32 23.59 -22 36 20.5   -3.85   0.99 1.53030646066456  -7.0135716  22.8493 /T  31.9096
 2019-Nov-11 00:00     16 37 41.64 -22 49 34.3   -3.85   1.00 1.52623945623356  -7.0691498  23.0875 /T  32.2499
 2019-Nov-12 00:00     16 43 00.67 -23 02 08.6   -3.85   1.00 1.52214051839967  -7.1241466  23.3252 /T  32.5901
 2019-Nov-13 00:00     16 48 20.62 -23 14 02.7   -3.85   1.00 1.51800996814825  -7.1786160  23.5623 /T  32.9303
```

Abbildung 10 - Ephemeriden Daten der Venus von der NASA Internetseite

Nach Studium der Daten, welche nichts an den eigentlichen Ellipsenbahnen ändern, sondern lediglich die Position eines Planeten zu einem bestimmten Zeitpunkt am Himmel definiert, haben diese Ephemeriden auf das Ergebnis keinen Einfluss, da sich die Figur sowieso bildet, weil es nur eine Frage der Zeit ist, bis sich die beiden Planeten, welche wir in Beziehung setzen wollen, in einer bestimmten (angenommenen) Positionen befinden. So wurde festgelegt, dass die Ausgangsposition, von der aus alle Berechnungen starten, in der Sonne/Planet 1/Planet 2 Konjunktion im entferntesten Punkt beginnt, was bedeutet, dass die Y-Position für beide Planeten Null ist ($y_1=0$, $y_2=0$) [12]. Somit beginnen wir die Berechnungen für beide Planeten in der Aphel Position bei y=0. Es ist leicht verständlich, dass diese Konstellation, je nach dem Wert des grossen Halbachsenabstandes der Planetenbahnellipse in wenigen (bei den inneren bzw. äusseren Planetenkombinationen) eintritt. Kombiniert man einen inneren Planeten (z.B. Merkur) mit einem äusseren Planeten (z.B. Neptun), so kann dies auch hundert und mehr Jahre dauern. Fakt ist aber, dass die angenommene Konjunktionsposition immer eintritt und somit

[12] y-Position von Planet 1 und Planet 2, z.B. Erde und Venus

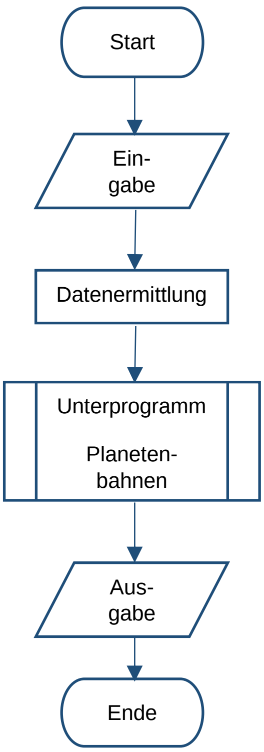
für den Nachweis einer evtl. vorhandenen höheren Ordnung, irgendeine Ausgangsposition angenommen werden kann. Wäre dies nicht so, so könnte eine höhere Ordnung nur für einen bestimmten Zeitraum gelten, was beweist, dass keine höhere Ordnung existiert.

Damit sind alle relevanten Parameter soweit geklärt, so dass mit der Programmausarbeitung begonnen werden kann.

8. Programmierung in TigerJython

In diesem Kapitel gehen wir nun vertieft in die notwendigen Softwarefunktionen ein, welche für die graphische Darstellung der Planetenbeziehungen notwendig sind. Die Programmierung erfolgt in der Programmiersprache TigerJython, da ich in der Vergangenheit diese bereits benutzt habe und entsprechende Kenntnisse vorliegen. Das Programm kann ebenfalls in jeder anderen Programmiersprache realisiert werden, z.B. Python, Java, VisualBasic, etc.

Das Planetenbahn-Visualisierungsprogramm folgt dem nachfolgend aufgeführten Programmkonzept.



Aufrufen des Programms in TigerJython

Auswahl der beiden Planeten, Farbe, Anzeige der Ellipsenbahnen, Auswahl Raumgerade bzw. Mittelpunkt und die Legende einstellen

Das Programm sucht sich auf Basis der Eingabe die notwendigen Planetendaten und Einstellungen aus Datenlisten/Datenmatrizen

Berechnen der Planetenpositionen und der Gestirnmittelpunktverbindungslinie

Ausgabe der Gestirnmittelpunktverbindungslinie bzw. Mittelpunkt der ausgewählten Planeten in der gewählten Farbe und Ausgabe als Ausdruck zur weiteren Verwendung

Schliessen des Programms

Abbildung 11 - Programmablaufplan

Die einzelnen Bereiche und Funktionen werden anschliessend genauer anhand des reellen Programmiercodes aufgeschlüsselt und detailliert erklärt;

8.1. Das Vorprogramm

```python
from gpanel import *
from math import *

makeGPanel("ErdeVenusBlume",-150, 150, -150, 150)
#Erde
setColor("blue")
circle(149.60)
#Venus
setColor("green")
circle(108.21)

def RaumgeradeEV():
    t=974           #Anzahl 3-Tage Schritte (hier 8 Jahre)
    we=0.051606         #Winkel Erde in Bogenmass
    wv=0.083887         #Winkel Venus in Bogenmass
    for x in range(1,t,1):
        lineWidth(1)
        setColor("red")
        xErde = (math.cos(we*x))*149.60
        yErde = (math.sin(we*x))*149.60
        xVenus = (math.cos(wv*x))*108.21
        yVenus = (math.sin(wv*x))*108.21
        line(xErde, yErde, xVenus, yVenus)

RaumgeradeEV()
```

Abbildung 12 - Quellcode des Vorprogrammes

Mit diesem Kurzprogramm wurde versucht die Planetenbewegungen der Ellipsenbahnen graphisch darzustellen, um sich einen ersten Überblick zu verschaffen, ob überhaupt ein Muster entsteht. Das Resultat wurde in das Konzept eingearbeitet und als Basis für die Maturaarbeit hergenommen. Als erste Planetenkombination wurde die ‚Erde-Venus' Blume ausgewählt, um zu sehen ob das Programm zum selben Ergebnis gelangt wie im Internet publiziert ist.

Im ersten Teil des Programmes werden alle notwendigen Daten importiert bzw. festgelegt. Dies sind die beiden mittleren Ellipsenachsenradien, sowie die beiden Umlaufzeiten in Tagen bzw. deren Winkelbetrag (im Bogenmass) im vorgegebenen Zeitintervall (hier 3 Tage). Der nachfolgende Programmteil berechnet die Planetenposition auf der Umlaufbahn zu bestimmten Zeitpunkten. Anschliessend werden die beiden Planetenposition im GPanel durch eine Verbindungsgerade

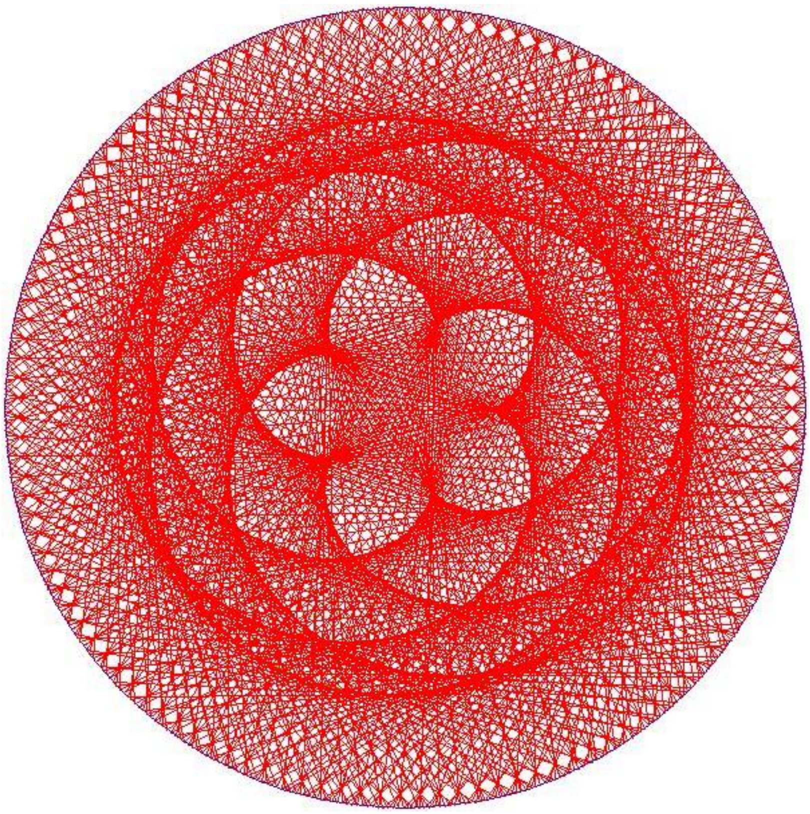


Abbildung 13 - Erde-Venus Blume des Vorprogrammes eingezeichnet. Mit der for-Schleife werden nun alle Zeitintervalle abgearbeitet und das Resultat in das GPanel gezeichnet, so dass sich eine Bewegungsgraphik über den festgelegten Zeitraum (974 Schritte à 3 Tage) ausbildet. Die vorhergehende Graphik (Abbildung 13) gibt das resultierende Bewegungsmuster wider, welches mit dem oben beschriebenen Programm erhalten wurde. Wie man sehen kann, bildet sich tatsächlich ein Blumenmuster heraus, welches dem Internetmuster 1:1 gleicht.

8.2. Die finale Software - Lösung

Nachdem klar war, dass ein Blumenmuster entsteht, wurde mit der Planung des ‚eigentlichen' Programms begonnen, bei dem die Funktionalität integriert wurde, um universell alle zu untersuchenden Kombinationen gegeneinander in Beziehung zu setzen und graphisch auszugeben. Das Programm soll alle möglichen Planetenkombinationen unseres Sonnensystems berücksichtigen, damit breit geforscht werden kann.

```
15   class program:
16
17 >     def printGPanel(self, event):···
20
21 >     def clearGPanel(self,event):···
23
24 >     def Legende(self, selectedIndex1, selectedIndex2):···
30
31 >     def Winkel1(self, InputWinkel1, TagesSchritte):···
34
35 >     def Winkel2(self, InputWinkel2, TagesSchritte):···
38
39       def Raumgerade(self, AnzahlTagesSchritte, W1, W2, Planet1EllipsenseiteAneu,
   >     Planet1EllipsenseiteBneu, Planet2EllipsenseiteAneu, Planet2EllipsenseiteBneu):···
47
48       def Mittelpunkt(self, AnzahlTagesSchritte, W1, W2, Planet1EllipsenseiteAneu,
   >     Planet1EllipsenseiteBneu, Planet2EllipsenseiteAneu, Planet2EllipsenseiteBneu):···
54
55 >     def actionCallback(self,event):···
128
129 >    def __init__(self):···
```

Abbildung 14 - Klasse des Programmes mit den Definitionen

Die Planeten werden via ComboBoxen ausgewählt und zusätzlich kann die Auswahl der Farbe, sowie die über RadioButtons die Art der zu zeichnenden Linien (Verbindungslinie der beiden Planeten, Ellipsenbahn, usw.) bzw. Mittelpunkt (Mittelpunkt der Verbindungslinie beider Planeten) festgelegt werden.

Das Programm wurde als Klasse definiert. Die Klasse wird einerseits für eine bessere Strukturierung verwendet. Dabei spricht man von objektorientierter Programmierung. Klassen besitzen bestimmte Eigenschaften, die Attribute und bestimmte Funktionen, die Methoden. Der Parameter «*self*» dient dazu, dass auf die Methoden und Attribute zugegriffen werden kann. Ohne diesen Parameter könnte man nicht auf den Inhalt der ComboBoxen zugreifen. Die Methoden sind in den Zeilen 17 bis 127 festgelegt und in der Zeile 169 wird die Klasse «*program*» als eigenständiges Programm ausgeführt. Tiefer in die Programmiertheorie bzw. –mathematik einzusteigen macht im Zuge der Maturaarbeit keinen Sinn, da dies zu theoretisch wird und keinen Beitrag zum Verständnis der Programmfunktion leistet.

Mit diesem Programmdesign wurde eine maximale Flexibilität offen gehalten. Zukünftige Erweiterungen können so einfach integriert werden, ohne dass der Programmkörper geändert werden muss.

8.2.1. Datenbanken und Rohdaten

```
from math import *
from javax.swing import *
from gpanel import *
```

Abbildung 15 - Import-Datenbanken

Im ersten Teil des Programms werden alle notwendigen (‚Programm') Datenbanken importiert. Dabei muss auf die Reihenfolge geachtet werden. Es ist wichtig, dass die Datenbank «*gpanel*» als letztes importiert wird, denn die Datenbank «*javax.swing*» beinhaltet ebenfalls einige Befehle, die den gleichen Namen tragen, jedoch nicht die identische Funktionalität bieten und somit zu Fehlern führen können. Hier muss genau recherchiert werden welche Funktionen in den importierten Programmdatenbanken enthalten sind, andernfalls führt dies zu Programmfehlern deren Lösung zu schlaflosen Nächten führen können. Der nachfolgende Abschnitt zeigt beispielhaft solch eine Situation und dessen Lösung auf.

Wird das Python «*gpanel*» vor der Datenbank «*javax.swing*» importieren, so tritt beim Ausführen des Programms (vorausgesetzt die Checkbox «*Legende*» wurde angeklickt) der nachfolgende Fehler auf; «*TypeError: 'javapackage' object is not callable*». Der Fehler wird durch den Befehl «*text()* aufgerufen. Dieser Fehler bedeutet, dass das Python Objekt, so wie im Programm verwendet, nicht aufgerufen werden kann, weil eine weitere Datenbank ebenfalls den Befehl enthält und somit ein Fehler auftritt. Deshalb ist es wichtig, die Reihenfolge der importierten Datenbanken eindeutig festzulegen. Dies setzt aber auch voraus, dass tiefere Kenntnisse über diese Programmdatenbanken und deren Inhalt vorhanden ist.

```
5   # P = Planeten; F = Farben; a = Ellipsenhalbachse a; b = Ellipsenhalbachse b; t = Umlaufzeit;
    g = GPanel-Koordinaten yMin, yMax, xMin, xMax
6   P = ["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus", "Neptune"]#Planeten:
    Mercury(0), Venus(1), Earth(2), Mars(3), Jupiter(4), Saturn(5), Uranus(6), Neptune(7)
7   F = ["Red", "Orange", "Yellow", "Lime", "Green", "Cyan", "Blue", "Magenta","Purple", "Black",
    "Gold"]#Farben: red(0), orange(1), yellow(2), lime(3), green(4), cyan(5), blue(6), magenta(7),
    purple(8), black(9)
8   a = [57.91, 108.21, 149.60, 227.92, 778.57, 1433.53, 2872.46, 4495.06]#(km*10^6)
9   b = [56.67, 108.21, 149.58, 226.92, 777.64, 1431.24, 2869.46, 4494.77]#(km*10^6)
10  t = [87.969, 224.701, 365.256, 686.980, 4332.589, 10759.22, 30685.4, 60189.0]#(Tage)
11  TagesSchritteMatrix = [[0,3,3,3,5,10,30,50],[3,0,3,5,10,10,15,50],[3,3,0,5,10,10,20,50],[3,5,5,
    0,10,10,50,50],[5,10,10,10,0,20,50,100],[10,10,10,10,20,0,50,300],[30,15,20,50,50,50,0,550],
    [50,50,50,50,100,300,550,0]]#(Tage)
12  AnzahlTagesSchritteMatrix = [[0,680,850,1150,867,1075,1030,1205],[680,0,974,2350,435,1080,2047,
    1205],[850,974,0,1120,438,1095,1535,1205],[1150,2350,1120,0,1305,1100,620,1205],[867,435,438,
    1305,0,1085,7350,607],[1075,1080,1095,1100,1085,0,4305,1000],[1030,2047,1535,620,7350,4305,0,
    2800],[1205,1205,1205,1205,607,1000,2800,0]]#(Tage)
13  g = 5000
```

Abbildung 16 - Primärdaten

In den Zeilen 6 bis 13 werden alle Matrizen, Listen und Konstanten festgelegt bzw. definiert. Diese Informationen bilden die Primärdaten der Planeten und aller weiteren Programmparameter, welche in der weiteren Ausführung berücksichtigt werden müssen.

Die Liste «*P*» beinhaltet alle Planeten unseres Sonnensystems, geordnet vom Sonnennächsten bis zum Sonnenferntesten Gestirn.

Die Liste «*F*» beinhaltet, die mögliche Auswahl an Farben, in denen die Linien/Punkte im GPanel gezeichnet werden.

Die Listen «*a*», «*b*» und «*t*» legen die Ellipsenhalbachsen a und b (in 10^6 km) fest, sowie die Umlaufzeit t der Planeten in Tagen. Die Listen sind vom Sonnennächsten bis zum Sonnenferntesten Planet geordnet.

Die Matrix «*TagesSchritteMatrix*» enthält die Tagesschritte in denen gerechnet bzw. gezeichnet wird für alle Planetenkombination.

Die Matrix «*AnzahlTagesSchritteMatrix*» beinhaltet die Zeitspanne, über die die Blume gezeichnet wird. Wie bei der Matrix «*TagesSchritteMatrix*» wurden hierbei alle möglichen Planetenkombinationen berücksichtigt.

Der Parameter «*g*» steht für die Grösse des der x,y bzw. –x,-y Achsen des GPanel-Koordinatensystems.

8.2.2. Der Konstruktor

Der Konstruktor ist eine Methode, die nicht aufgerufen werden kann. Er wird beim ausführen des Programmcodes der Klasse automatisch aufgerufen. Im Konstruktor werden die beiden Comboboxen definiert, welche die Planeten die gegeneinander in Beziehung gestellt werden sollen, ausgewählt. Diese ComboBoxen werden mit den Daten der Liste «*P*» (Zeile 6) vorgefüllt, damit diese Daten zur Auswahl zur Verfügung stehen.

Die dritte Combobox ist für die auswählbare Farbe zur Darstellung der Graphik zuständig. Sie wird mit den Datenelementen der Liste «*F*» (Zeile 7) gefüllt.

```
    def __init__(self):

        self.ComboBox1 = JComboBox(P)
        self.ComboBox2 = JComboBox(P)
        self.ComboBoxFarbe = JComboBox(F)

        self.CheckBoxE = JCheckBox("Ellipsen")
        self.CheckBoxL = JCheckBox("Legende")

        self.RadioBtnR = JRadioButton("Raumgerade")
        self.RadioBtnM = JRadioButton("Mittelpunkt")
        RadioButtonGroup = ButtonGroup()
        RadioButtonGroup.add(self.RadioBtnR)
        RadioButtonGroup.add(self.RadioBtnM)

        OkButton = JButton("OK", actionPerformed = self.actionCallback)
        ClearButton = JButton ("Clear", actionPerformed = self.clearGPanel)
        PrintButton = JButton ("Print", actionPerformed = self.printGPanel)
```

Abbildung 17 - Buttons des Konstruktors

In Zeile 135 und 136 werden die Checkboxen definiert. Die erste Checkbox bekommt den Namen «*Ellipsen*» und legt fest, ob die Planetenbahnen zusätzlich zu den Raumgeraden (Verbindungslinie bzw. Mittelpunkt der Verbindungslinie) mitgezeichnet werden sollen.

Die zweite CheckBox hat den Namen «*Legende*», hier wird festgelegt, ob die Planetenbahnen und die wichtigsten Parameter als Legende angezeigt werden sollen.

Als nächstes werden die Radiobuttons definiert. In Zeile 138 und 139 werden die zwei Radiobuttons erstellt, die den Namen «*Raumgerade*» und «*Mittelpunkt*» erhalten. Anschliessend muss noch eine Radiobutton-Gruppe erstellt werden, in die die zwei Radiobuttons eingebunden sind. Die Gruppendefinition führt dazu, dass nicht beide Radiobuttons gleichzeitig ausgewählt werden können, so dass zwischen den beiden Optionen gewählt werden muss.

Dies macht Sinn, wenn sich zwei Optionen ausschliessen und somit zur Vermeidung einer Falschauswahl des Benutzers, Vorkehrungen eingeleitet werden welche verhindern, dass das Programm ungewollte Aktionen ausführt.

```
148        MenuBar = JMenuBar()
149        MenuBar.add(self.ComboBox1)
150        MenuBar.add(self.ComboBox2)
151        MenuBar.add(self.ComboBoxFarbe)
152        MenuBar.add(self.CheckBoxE)
153        MenuBar.add(self.CheckBoxL)
154        MenuBar.add(self.RadioBtnR)
155        MenuBar.add(self.RadioBtnM)
156        MenuBar.add(OkButton)
157        MenuBar.add(ClearButton)
158        MenuBar.add(PrintButton)
```

Abbildung 18 - MenuBar des Konstruktors

In den Zeilen 144 bis 146 werden die Buttons erstellt. Der «*OkButton*» startet das Programm wobei alle ausgewählten Optionen ermittelt und ausgewertet werden. Der OkButton startet, wenn er betätigt wird, die Methode «*actionCallback*», die später noch erklärt wird. Der Befehl sieht dann wie folgt aus: «*OkButton = JButton(«OK», actionPerformed = self.actionCallback)*».

Funktional (im Hinblick auf die programmtechnische Arbeitsweise) identisch wird der Button «*ClearButton*» definiert, mit dem Unterschied, dass er den Namen «*Clear*» trägt und beim Betätigen die Methode «*clearGPanel*» ausführt und somit die Graphik, indem er sie mit weisser Farbe überzeichnet, löscht.

Der Button «*PrintButton*» in Zeile 146 trägt den Namen «*Print*» und beim Betätigen wird die Methode «*printGPanel*» ausgeführt, welche das Druckfenster öffnet.

Hier kann man auch weitere Druckprogramme als Druckertreiber einbinden, welche eine Graphikdatei generieren, wie zum Beispiel gif, jpg, png oder weitere Formate. Im Internet kann man hierzu lizenzfreie Programme finden, welche einfach in das Betriebssystem eingebunden werden können.

In Zeile 148 wird die MenuBar erstellt. Alle zuvor genannten Elemente wurden in diese (Zeilen 131 bis 146) MenuBar integriert und somit die gesamte Funktionalität bereitgestellt.

In Zeile 160 wird das GPanel generiert: «*makeGPanel(MenuBar, -g, g, -g, g)*». Der erste Ausdruck in der Klammer fügt die MenuBar dem GPanel hinzu. Die vier darauffolgenden Ausdrücke «*-g, g, -g, g*», definieren das Koordinatensystem und deren Grösse der einzelnen Achsen (x-Min, x-Max, y-Min, y-Max). Danach wird der Name des Fensters von «*GPanel*» auf «*Kepler's Sphaerengeometrie*» geändert.

```
160        makeGPanel(MenuBar, -g, g, -g, g)
161
162        title("Kepler's Sphaerengeometrie")
163        resizable(False)
164        windowSize(1000, 1000)
165        windowPosition(10, 10)
166
167        validate()
```

Abbildung 19 - GPanel Definition

Der Befehl: «*resizable(False)*» setzt die Grössenänderung des Fensters auf *false*. Dies verhindert, dass das GPanel in der Grösse mit der Maus verändert werden kann. Anschliessend erfolgt die Grösseneinstellung des Fensters in Pixeln. Mit «*windowPosition(10, 10)*» wird das Fenster entsprechend auf dem Monitor positioniert, so dass das Fenster auch sichtbar ist.

Mit dem Befehl «*validate()*» aus Zeile 167 wird schlussendlich das Fenster mit den hinzugefügten Komponenten neu aufgebaut, anschliessend kann der Benutzer seine Auswahl mit der Maus tätigen.

8.2.3. Methoden

Die «actionCallback» Methode

```
55      def actionCallback(self,event):
56
57          selectedIndex1 = self.ComboBox1.selectedIndex
58          selectedIndex2 = self.ComboBox2.selectedIndex
59          selectedC = self.ComboBoxFarbe.selectedIndex
```

Abbildung 21 - "actionCallback" Methode Selektion

Beim Drücken des «*OkButtons*» wird die Methode «*actionCallback*» (Zeile 55 bis 127) ausgeführt. Als erstes werden in dieser Methode alle Indizes der ausgewählten Elemente der Comboboxen als Variablen festgelegt bzw. definiert. Weil die Matrizen, wie die Comboboxen, von den Sonnennächsten bis zu den Sonnenferntesten Planeten geordnet sind,

```
61          TagesSchritte = TagesSchritteMatrix[selectedIndex1][selectedIndex2]
62          AnzahlTagesSchritte = AnzahlTagesSchritteMatrix[selectedIndex1][selectedIndex2]
63
64          #Planet 1 Daten
65          InputWinkel1 = t[selectedIndex1]
66          W1 = self.Winkel1(InputWinkel1, TagesSchritte)
67          Planet1EllipsenseiteA = a[selectedIndex1]
68          Planet1EllipsenseiteB = b[selectedIndex1]
69
70          #Planet 2 Daten
71          InputWinkel2 = t[selectedIndex2]
72          W2 = self.Winkel2(InputWinkel2, TagesSchritte)
73          Planet2EllipsenseiteA = a[selectedIndex2]
74          Planet2EllipsenseiteB = b[selectedIndex2]
```

Abbildung 20 - "actionCallback" Methode Variablen

kann mit Einsetzen der Indizes der Comboboxen in die Matrix die richtige Zahl aus der Matrix abgerufen werden. Dies wird für die beiden Matrizen in den Zeilen 11 und 12 angewendet, welche den zugehörigen Wert aus der Matrix abruft. Im nächsten Schritt, Zeile 64 bis 74, werden die benötigten Rohdaten aus den Listen der Zeilen 8 bis 10 entnommen und ebenfalls als Variablen definiert.

Der Winkel, der für das Zeichnen des Bildes benötigt wird, muss mit den Methoden «*Winkel1*» und «*Winkel2*» berechnet werden. Diese werden anschliessend mittels gegebener Umlaufzeit

```
112         if self.CheckBoxE.isSelected():
113             lineWidth(1)
114             ellipse(Planet1EllipsenseiteAneu, Planet1EllipsenseiteBneu)
115             ellipse(Planet2EllipsenseiteAneu, Planet2EllipsenseiteBneu)
```

Abbildung 22 - "actionCallback" Methode Ellipsen-CheckBox

aus der Liste «*t*», als «*InputWinkel1*» und «*InputWinkel2*» definiert. Als Ergebnis resultieren die Variablen «*W1*» und «*W2*».

```
76          if Planet1EllipsenseiteA >= Planet2EllipsenseiteA:
77              x = g/Planet1EllipsenseiteA
78              Planet1EllipsenseiteAneu = Planet1EllipsenseiteA*x
79              Planet1EllipsenseiteBneu = Planet1EllipsenseiteB*x
80              Planet2EllipsenseiteAneu = Planet2EllipsenseiteA*x
81              Planet2EllipsenseiteBneu = Planet2EllipsenseiteB*x
82          else:
83              x = g/Planet2EllipsenseiteA
84              Planet1EllipsenseiteAneu = Planet1EllipsenseiteA*x
85              Planet1EllipsenseiteBneu = Planet1EllipsenseiteB*x
86              Planet2EllipsenseiteAneu = Planet2EllipsenseiteA*x
87              Planet2EllipsenseiteBneu = Planet2EllipsenseiteB*x
```

Abbildung 23 - "actionCallback" Methode Skalierung

Zeile 76 bis 87 berechnen die neuen Ellipsenbahnpositionen. Das ist nötig, weil das Koordinatensystem eine fixe Grösse haben muss und die Planeten unterschiedlich grosse Ellipsenbahnen ziehen. Zeile 76 vergleicht nun die beiden Ellipsenbahnen und bestimmt welche Bahn aussen und welche innen zu liegen kommt. Die Grösse der Graphik wird anschliessend skaliert, so dass die Grösse der Figur, welche auf dem GPanel bzw. dem Ausdruck (Datei) angezeigt wird immer dieselbe Grösse hat. Somit ist das Bild auf die Grösse des Koordinatensystems skaliert und passt dadurch immer in das angezeigte GPanel bzw. dessen Koordinatensystem.

In Zeile 89 bis 110 wird die Farbe, abhängig von der Auswahl der Combobox, eingestellt. Wenn die Checkbox «*Ellipsen*» ausgewählt wurde, werden in Zeile 113 bis 115 die Ellipsenbahnen der ausgewählten Planeten mitgezeichnet.

In Zeile 117 bis 122 bestimmt das Programm, welche der zwei Radiobuttons («*Raumgerade*» oder «*Mittelpunkt*») ausgewählt wurde und führt, bezogen auf die Auswahl, die dazugehörige Methode aus. Zum Beispiel wurde der Radiobutton «Raumgerade» ausgewählt, dann wird die Methode «Raumgerade» (Zeile 39 bis 46) aufgerufen.

```
        if self.RadioBtnR.isSelected():
            self.Raumgerade(AnzahlTagesSchritte, W1, W2, Planet1EllipsenseiteAneu,
            Planet1EllipsenseiteBneu, Planet2EllipsenseiteAneu, Planet2EllipsenseiteBneu)
        elif self.RadioBtnM.isSelected():
            self.Mittelpunkt(AnzahlTagesSchritte, W1, W2, Planet1EllipsenseiteAneu,
            Planet1EllipsenseiteBneu, Planet2EllipsenseiteAneu, Planet2EllipsenseiteBneu)
        else:
            print "Fehler"
```

Abbildung 24 - "actionCallback" Methode Graphik-Auswahl

Falls keine der zwei Radiobuttons ausgewählt wurden, wird in der Ausgabe der Begriff «*Fehler*» ausgegeben.

Nachdem das eigentliche Bild gezeichnet ist, wird in Zeile 124 festgestellt, ob die Checkbox «*Legende*» ausgewählt wurde. Ist dies der Fall, wird die Methode «*Legende*» (Zeile 24 bis 29) ausgeführt und der Graphik eine Legende mit den relevanten Informationen, ausgegeben.

Die «printGPanel» Methode

```
17    def printGPanel(self, event):
18        printScreen(0.8)
19        dispose()
```

Abbildung 25 - "printGPanel" Methode

Bei Betätigung des Printbutton wird die Methode «*printGPanel*» ausgeführt (Zeile 17 bis 19), dabei öffnet sich das Druckfenster, um den Drucker bzw. das Dateiformat (z.B. pdf) auszuwählen. Falls «*print to PDF*» ausgewählt wird, öffnet sich ein weiteres Fenster, das den Speicherort und den Dateinamen abfragt. Das Bild wird in der Skalierung 0.8 «*printScreen(0.8)*» ausgegeben, dies gewährleistet, dass keine Teile abgeschnitten werden. Je nach Druckertreiben können Bilder in einer Auflösung bis zu 2400 dpi gespeichert werden.

Die «clearGPanel» Methode

```
21    def clearGPanel(self,event):
22        clear()
```

Abbildung 26 - "clearGPanel" Methode

In Zeile 21 und 22 wird die Methode «*clearGPanel*» definiert. Dieser Befehl «*clear()*» löscht das GPanel.

Die «Legende» Methode

```
24    def Legende(self, selectedIndex1, selectedIndex2):
25        AnzahlJahresSchritte = AnzahlTagesSchritteMatrix[selectedIndex1][selectedIndex2]
          *TagesSchritteMatrix[selectedIndex1][selectedIndex2]/t[2]
26        setColor("black")
27        text(-4950, -4500, str(P[selectedIndex1] + ", " + P[selectedIndex2]))
28        text(-4950, -4650, str(TagesSchritteMatrix[selectedIndex1][selectedIndex2]) + "
          Tages-Schritte")
29        text(-4950, -4800, str(round(AnzahlJahresSchritte, 1)) + " Jahre Zeitspanne")
```

Abbildung 27 - "Legende" Methode

Wenn die Checkbox «*Legende*» ausgewählt wurde, wird diese Methode (Zeile 24 bis 29) ausgeführt. Dabei braucht die Methode zwei Parameter, die in der Methode «*actionCallback*» definiert worden sind und einen Wert bekommen haben.

Diese Parameter sind «*selectedIndex1*» und «*selectedIndex2*». Im ersten Schritt der Methode wird die Zeitspanne in Jahre umgerechnet. Dazu multipliziert das Programm die zutreffende Zahl der Matrix «*AnzahlTagesSchritteMatrix*» mit der zutreffenden Zahl der Matrix «*TagesSchritteMatrix*» und teilt es dann durch die dritte Zahl der Liste «*t*», was der Erdumrundungszeit der Erde um die Sonne in Tagen entspricht. Man teilt durch diese Zahl, weil man als Resultat Erdenjahre benötigt. In Zeile 27 bis 29 wird die Legende im GPanel eingetragen.

Die «Winkel1», «Winkel2», «Raumgerade» und «Mittelpunkt» - Methoden

```
31    def Winkel1(self, InputWinkel1, TagesSchritte):
32        Winkel1def = 360/InputWinkel1*pi/180*TagesSchritte
33        return Winkel1def
34
35    def Winkel2(self, InputWinkel2, TagesSchritte):
36        Winkel2def = 360/InputWinkel2*pi/180*TagesSchritte
37        return Winkel2def
```

Abbildung 28 - "Winkel1" & "Winkel2" Methoden

Die «*Winkel1*» und «*Winkel2*» Methoden (Zeile 31 bis 33 und Zeile 35 bis 37) berechnen den Winkel in Bogenmass, mit dem das Programm die Bilder, bezogen auf die ausgewählten Planeten, zeichnet. Als Input braucht die Methode die Variabel «*InputWinkel1*» bzw. «*InputWinkel2*» und die Variable «*TagesSchritte*». Diese Variablen wurden in der Methode «*actionCallback*» definiert.

In Zeile 32 und 36 werden die Winkel berechnet und als Variabeln «*Winkel1def*» und «*Winkel2def*» abgespeichert. Dabei rechnet die Methode 360 geteilt durch den «*InputWinkel1*» oder «*InputWinkel2*» mal π (Zahl Pi) und geteilt durch 180 mal Anzahl «*TagesSchritte*». Im nächsten Schritt wird mit «*return*» der Wert der Variabeln «*Winkel1def*» bzw. «*Winkel2def*» dem Namen der Methode überschrieben. Somit hat zum Beispiel die Methode «*Winkel1*» den Wert von «*Winkel1def*».

```
39    def Raumgerade(self, AnzahlTagesSchritte, W1, W2, Planet1EllipsenseiteAneu,
      Planet1EllipsenseiteBneu, Planet2EllipsenseiteAneu, Planet2EllipsenseiteBneu):
40        lineWidth(1)
41        for x in range(1,AnzahlTagesSchritte,1):
42            xLineP1 = (math.cos(W1*x))*Planet1EllipsenseiteAneu
43            yLineP1 = (math.sin(W1*x))*Planet1EllipsenseiteBneu
44            xLineP2 = (math.cos(W2*x))*Planet2EllipsenseiteAneu
45            yLineP2 = (math.sin(W2*x))*Planet2EllipsenseiteBneu
46            line(xLineP1, yLineP1, xLineP2, yLineP2)
```

Abbildung 29 - "Raumgerade" Methode

Wenn der Radiobutton «*Raumgerade*» ausgewählt wurde, wird diese Methode (Zeile 39 bis 46) ausgeführt. Als Input braucht sie diese Variablen: «*AnzahlTagesSchritte, W1, W2, Pla-*

net1EllipsenseiteAneu, Planet1EllipsenseiteBneu, Planet2EllipsenseiteAneu, Planet2EllipsenseiteBneu». In Zeile 40 wird die Liniendicke auf eins gesetzt. Das Zeichnen der Blume in Zeile 41 bis 46 funktioniert nach dem gleichen Prinzip, wie im Vorprogramm. An erster Stelle in der Klammer der for-Schleife steht der Anfangswert, an zweiter Stelle der Endwert und an dritter Stelle die Grösse der Schritte, die das Programm beim Abarbeiten machen soll.

In den Zeilen 42 bis 45 werden die x- und y-Koordinaten der beiden Punkte zu einem bestimmten Zeitpunkt definiert und in Zeile 46 wird mit diesen Daten die Linie gezogen. Die x-Koordinaten werden mit dem Cosinus von W1 mal x und am Schluss mal die Länge der Ellipsenseite a berechnet. Die y-Koordinaten werden nach dem gleichen Prinzip aufbauend berechnet, nur dass nicht der Cosinus, sondern der Sinus zum Berechnen verwendet wird. Da x sich bei jedem Schritt der for-Schleife um eins erhöht wird der Winkel bei jedem Schritt um den Wert von «*W1*» bzw. «*W2*» grösser.

```
48      def Mittelpunkt(self, AnzahlTagesSchritte, W1, W2, Planet1EllipsenseiteAneu,
        Planet1EllipsenseiteBneu, Planet2EllipsenseiteAneu, Planet2EllipsenseiteBneu):
49          lineWidth(3)
50          for x in range(1,AnzahlTagesSchritte,1):
51              xPoint = (((math.cos(W1*x))*Planet1EllipsenseiteAneu)+((math.cos(W2*x))
                *Planet2EllipsenseiteAneu))/2
52              yPoint = (((math.sin(W1*x))*Planet1EllipsenseiteBneu)+((math.sin(W2*x))
                *Planet2EllipsenseiteBneu))/2
53              point(xPoint, yPoint)
```

Abbildung 30 - "Mittelpunkt" Methode

Wenn der Radiobutton «Mittelpunkt» ausgewählt wurde, wird diese Methode (Zeile 48 bis 53) aufgerufen. In Zeile 49 wird die Liniendicke auf drei gestellt. Die for-Schleife funktioniert genauso wie die der Methode «*Raumgerade*». Jedoch zeichnet diese Methode nicht eine Linie, sondern einen Punkt im Zentrum der Raumgerade. Die x-Koordinate des Punktes wird ausgerechnet indem man die x-Koordinate des ersten Planeten nimmt, diese mit der x-Koordinate des zweiten Planeten addiert und das Resultat durch zwei teilt. Die y-Koordinate wird auf die gleiche Art und Weise ausgerechnet und ins GPanel gezeichnet.

8.3. Programminterface

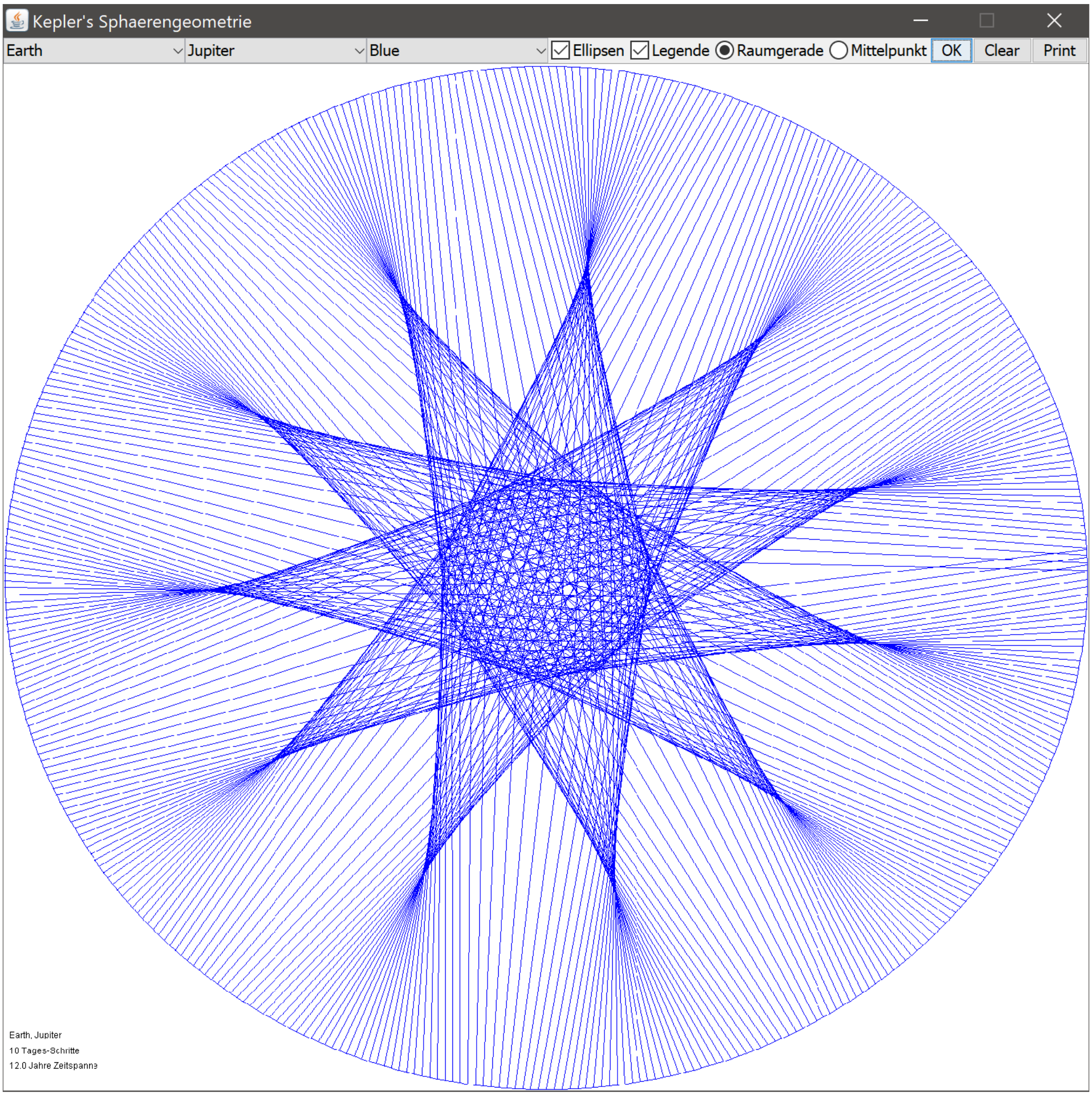


Abbildung 31 - das Programminterface

9. Formen aus Planetenkombinationen

In diesem Kapitel stellen wir die einzelnen Graphiken, welche aus den verschiedenen Planetenkombinationen erhalten wurden vor. Ausgewählt wurden hierzu nur Graphiken, welche auch eine Aussagekraft haben. In den einzelnen Unterkapiteln 9.1 bis 9.7 sind die Ergebnisse wiedergeben.

9.1. Venus-Erde Figur

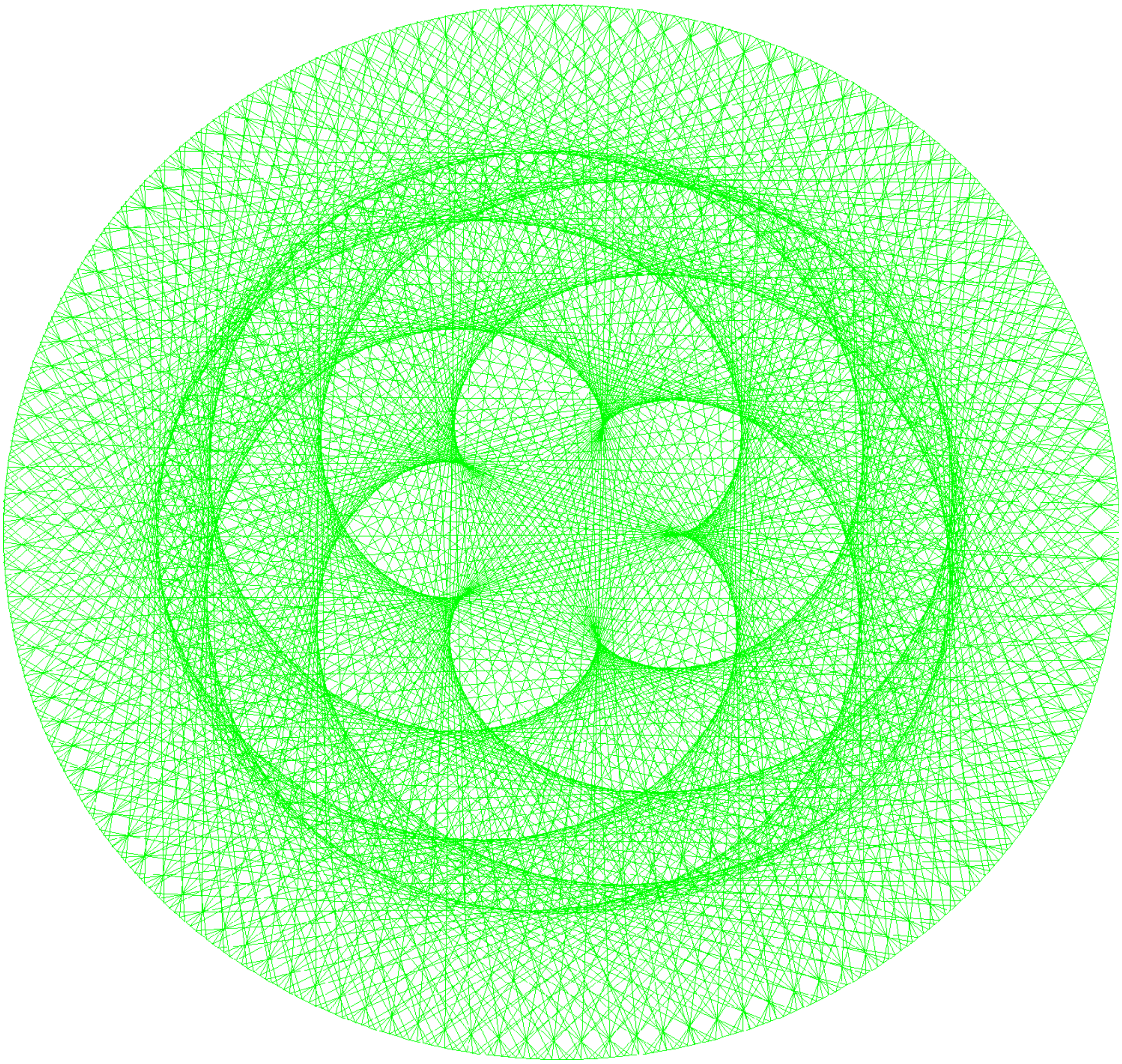


Abbildung 32 - Venus-Erde Graphik

Legende: Raumgerade, 3 Tagesschritte, 8 Jahre

Als erste Graphik ist das Ergebnis der Erde-Venus Verbindungslinien über einen Zeitraum von 8 Jahren in 3-Tagesschritten zu sehen. Die 8 Jahre haben sich empirisch ergeben, denn nur so war die resultierende Blume vollständig ausgebildet. Wie in den vorhergehenden Kapiteln zu sehen ist, kann man auch die Zeitperiode für diese Planetenkonstellation berechnen. Die Bestätigung, ob die Funktion nur ein Zufall ist, wurde nicht weiter untersucht.

9.2. Erde-Jupiter Figur

Abbildung 33 - Erde-Jupiter Graphik

Legende: Raumgerade, 10 Tagesschritte, 12 Jahre

Hier bildet sich ein Stern mit 11 Zacken heraus.

9.3. Jupiter-Saturn Figur

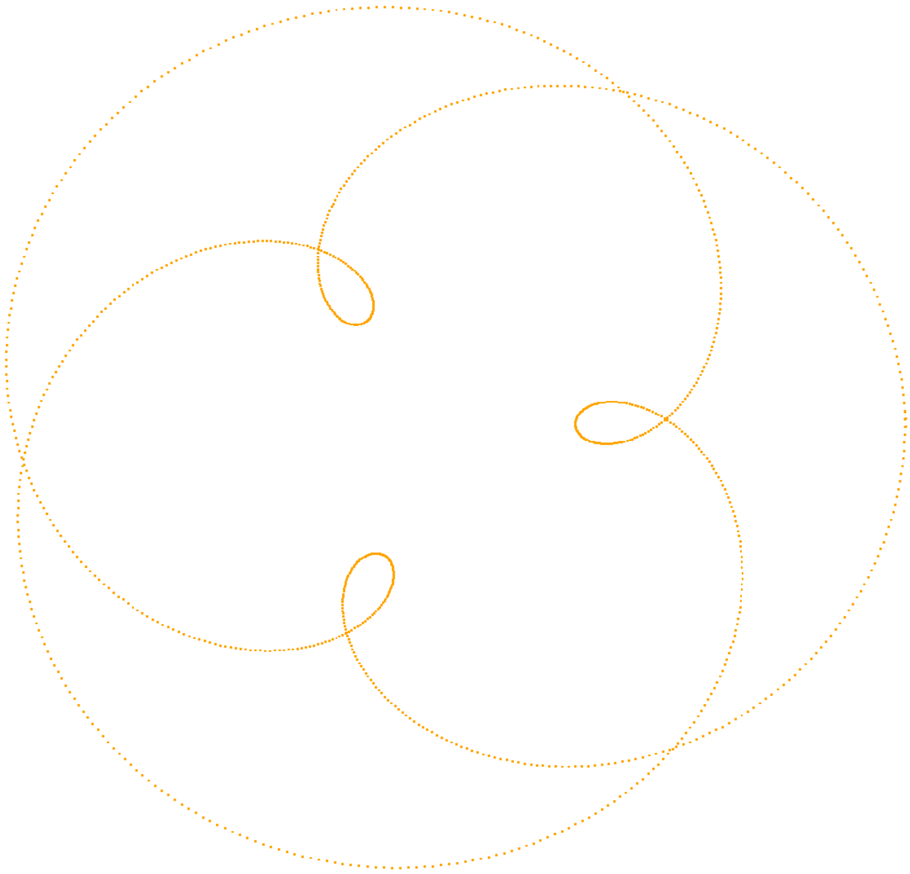


Abbildung 34 - Jupiter-Saturn Graphik

Legende: Schwerpunkt, 20 Tagessschritte, 59.4 Jahre

Hier kann nach der Simulation von ca. 60 Jahren die Bildung einer vorerst 3-blättrigen Blume erahnt werden.

9.4. Mars-Jupiter Figur

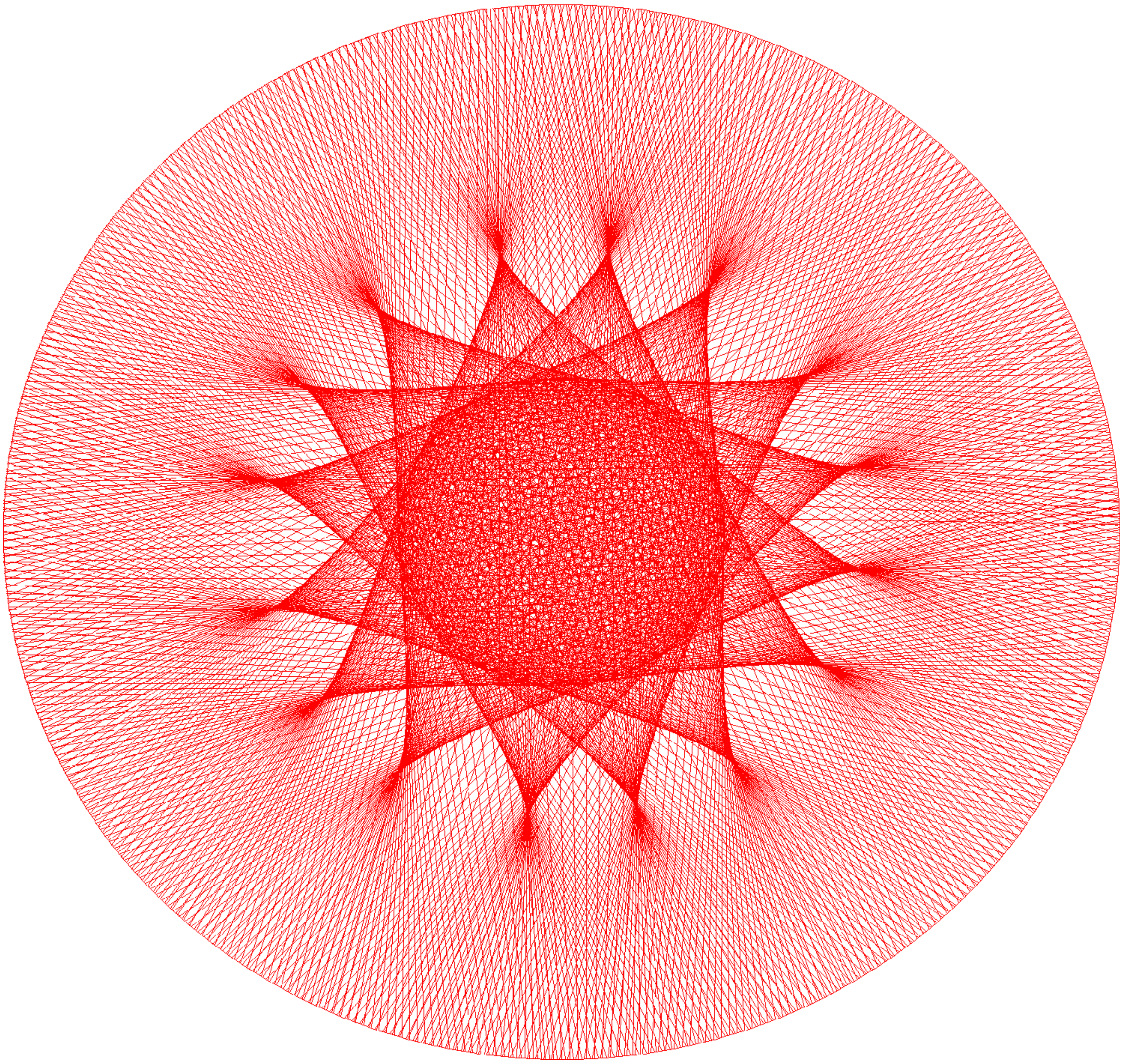


Abbildung 35 - Mars-Jupiter Graphik

Legende: Raumgerade, 10 Tagesschritte, 35.7 Jahre

Bei dieser Konstellation bildet sich ein Stern mit 16 Zacken aus.

9.5. Erde-Mars Figur

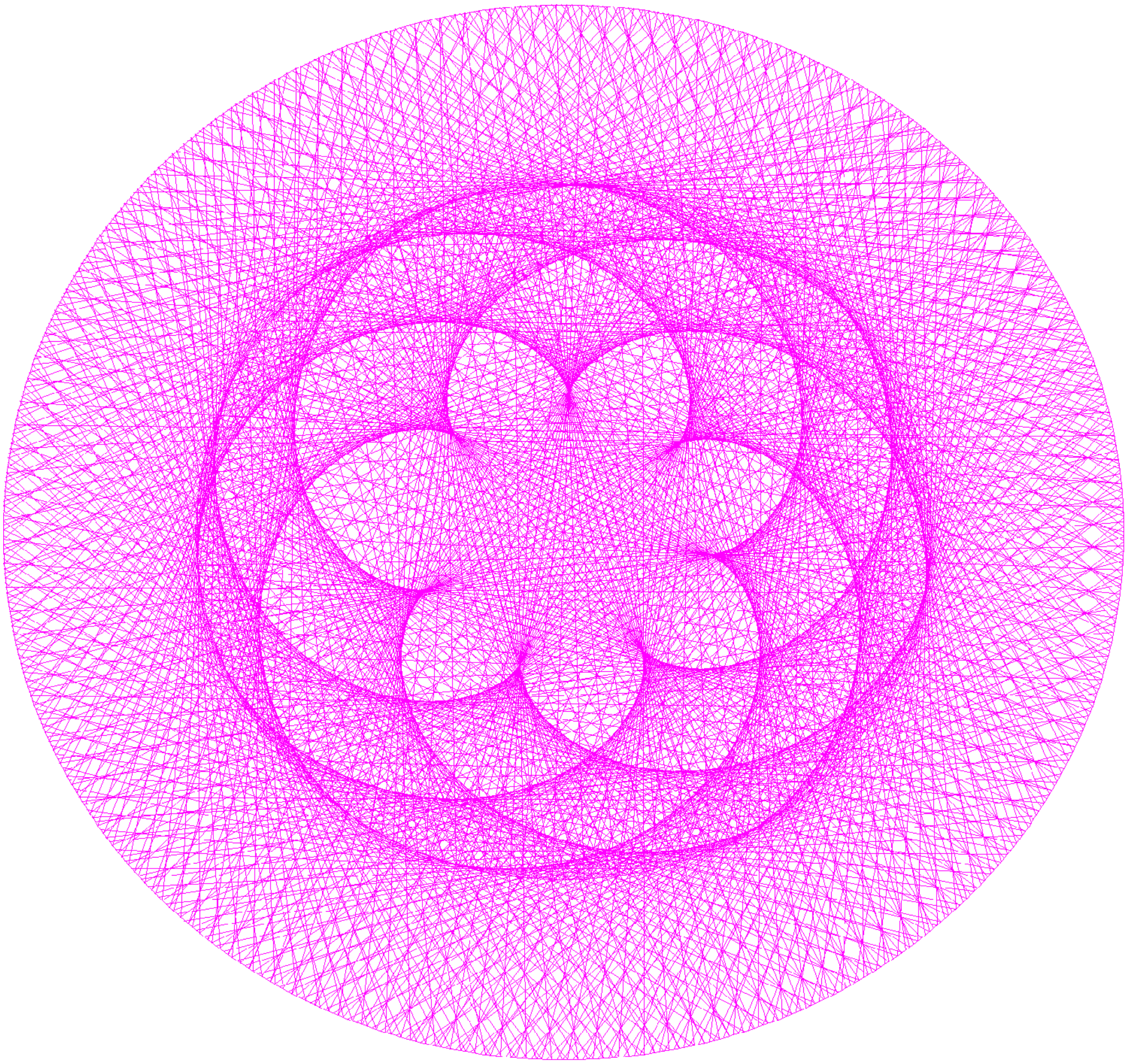


Abbildung 36 - Erde-Mars Graphik

Legende: Raumgerade, 5 Tagesschritte, 15.3 Jahre

Erde-Mars bilden nach ca. 15 Jahren eine 7-blättrige Blume aus, bei welcher ein Blatt deutlich grösser ausfällt.

9.6. Saturn-Uranus Figur

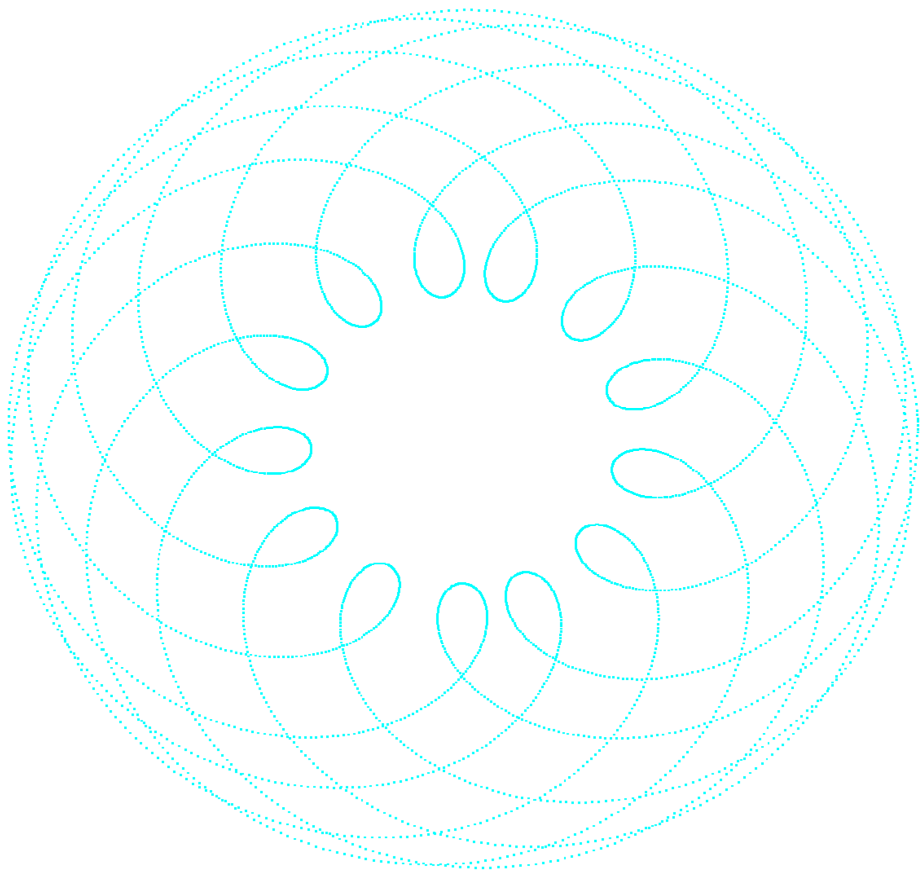


Abbildung 37 - Saturn-Uranus Graphik

Legende: Mittelpunkt, 50 Tagesschritte, 589.3 Jahre

Diese Figur kann nicht eindeutig zugeordnet werden. Die Darstellung der Raumgerade lässt leider, bedingt durch die vielen Geraden, welche im Laufe der fast 600 Jahre gezeichnet wurden, nichts Spezifisches erkennen.

9.7. Merkur-Venus Figur

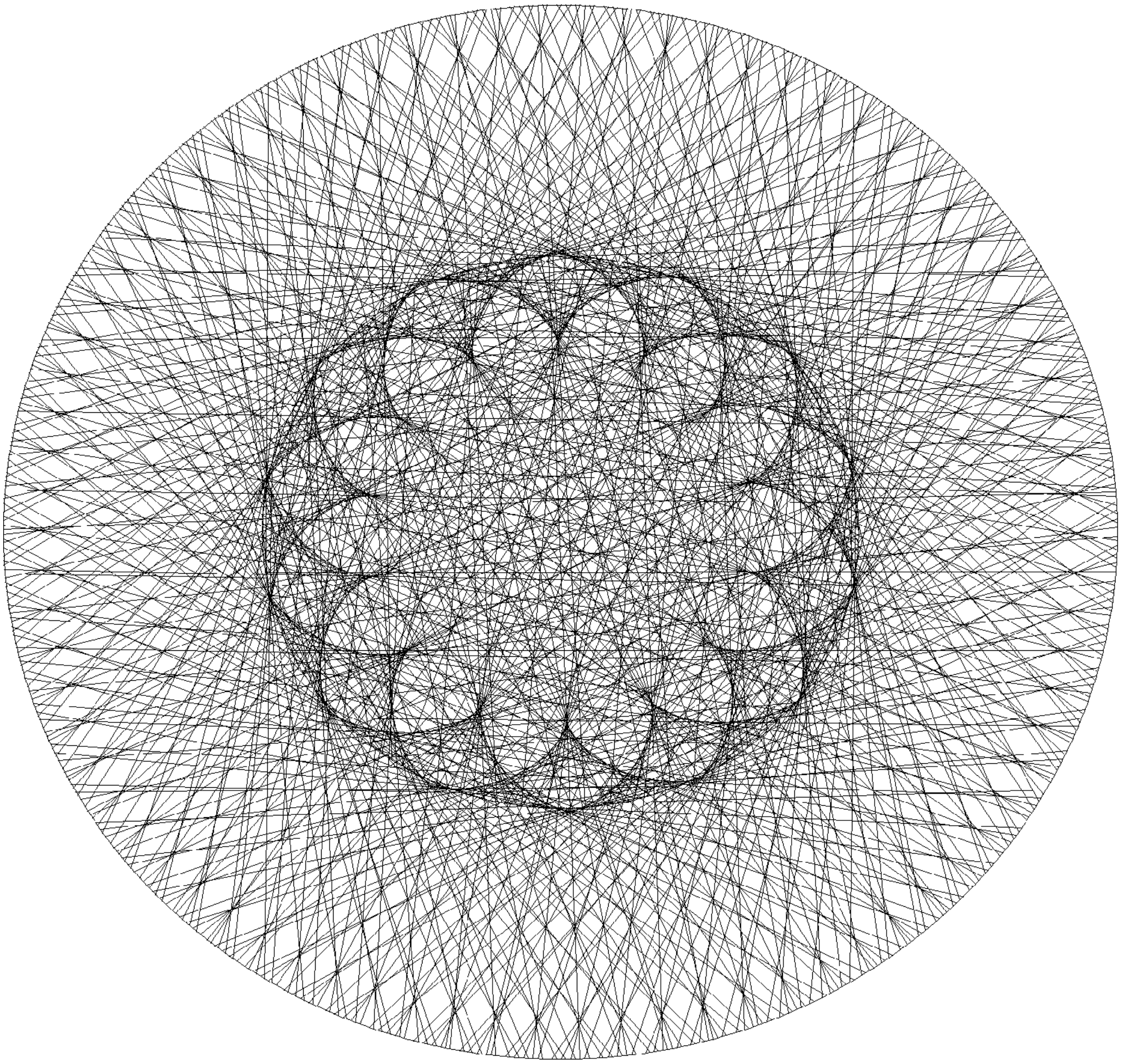


Abbildung 38 - Merkur-Venus Graphik

Legende: Raumgerade, 3 Tagesschritte, 5.6 Jahre

Merkur-Venus bildet eine 14-blättrige Blume aus, welche ebenfalls ‚fast' perfekt ausgebildet ist.

10. Schlussfolgerung/Resultat

Jede Planetenkonstellationsposition in unserem Sonnensystem ist einmalig und wiederholt sich nicht ein zweites Mal. Das heisst, wenn sich ein Planet auf demselben Längengrad mit der Sonne befindet, ist seine Deklination nie mehr identisch mit einer früheren Position, welche in Beziehung zu einem zweiten Gestirn steht. D.h. seine absolute Konstellationsposition nimmt der Planet in Bezug auf den zweiten Planeten und die Sonne (z.B. Sonne-Erde-Venus) nie mehr ein. Erstaunlich in diesem Zusammenhang ist die Tatsache, dass unser Sonnensystem, trotz der stetigen Änderungen nicht zerfällt und seit Millionen von Jahren stabil im Raum steht und noch Millionen von Jahren stehen wird.

Wie eindrücklich die Erde-Venus-/Erde-Mars-/Merkur-Venus-Blume zeigen, bilden sich faszinierende Figuren heraus, die einen in Erstaunen versetzen können. Viele weitere beeindruckende Figuren und Formen traten in Erscheinung, was einen zum Schluss kommen lässt, dass dahinter doch etwas verborgen zu sein scheint. Rein wissenschaftlich beurteilt, könnte es aber auch sein, dass durch die sich wiederholenden, regelmässigen Bewegungen der Planeten, auf ihren elliptischen Bahnen, zwei sich harmonisch verhaltende Objekte (hier die Planeten) gegeneinander in Beziehung gebracht werden und als Resultat daraus logischerweise auch ein harmonisches Bewegungsmuster auftritt.

Auf keinen Fall kann aus den erhaltenen Mustern auf eine höhere Ordnung geschlossen werden, dazu ist die Datenbasis für eine solche Schlussfolgerung zu gering. Nur weil man etwas gerne so sehen würde, eine Halbwahrheit zu konstruieren, um eine höhere Ordnung vermeintlich zu beweisen, wäre im höchsten Grad unwissenschaftlich. Deshalb lässt sich auf Basis der Bilder die Frage nicht abschliessend klären. Erstaunlich ist, dass die Blume und der Stern sehr häufig als Ergebnis auftreten, aber immer komplett unterschiedlich aussehen. So hat die Blume/Stern nie dieselbe Form und unterscheidet sich in der Blattanzahl bzw. der Anzahl der Zacken der Sternfiguren. Auch sind die berechneten Figuren nie 100% perfekt, so dass immer ein kleiner bzw. grösserer Winkelversatz zu beobachten ist, deshalb muss im Rahmen dieser Maturaarbeit leider gesagt werden, dass keine höhere Ordnung bestätigt werden konnte.

Auch scheint auf Basis der Resultate, welche ‚nur' einem beschränkten Untersuchungsumfang aufweisen, es unmöglich ist eine Beziehung in den Bewegungsmustern zu finden. Das Universum ist unendlich, so dass wir dies mit unseren beschränkten Verstand und Wissen ‚noch' nicht fassen können was da genau abläuft. Dazu bedarf es wahrscheinlich mehr als einer Maturaarbeit!

Das Zitat von John Locke (1632-1704) fasst dies in philosophische Worte;

*„Der beste Weg zur Wahrheit ist, die Dinge so zu betrachten, wie sie sind, und nicht so, wie wir folgern, dass sie zu sein hätten, wie wir sie uns vorstellen oder wie wir von anderen gelehrt wurden, sie uns vorzustellen." (*https://1000-zitate.de/autor/John+Locke/ [Stand: 26.11.2019])

11. Weiterführende Arbeiten und Ausblick

Hartmut Warm hat auf seiner Homepage diverse Figuren mit zwei Planeten und der Sonne in Konjunktion abgebildet, welche ebenfalls interessante Figuren/Graphiken zeigen. Hierzu könnte man nun auch eine Programmerweiterung ins Auge fassen, um auch die Dreierkombinationen Sonne und zwei weitere Planeten auszuwählen und gegeneinander in Konjunktionsposition bzw. als Dreiecksschwerpunkt in festen Zeitabständen darzustellen.

Um dem ‚Geheimnis' auf die Spur zu kommen, welche evtl. beweist, dass eine höhere Ordnung existiert und der Grund der entstandenen Blumen-Figuren ist, müsste eine Systematik gefunden werden, welche klar beweist nach welchen Regeln diese Figuren entstehen.

Die Komplexität unseres Sonnensystems und des Universums allgemein zu verstehen, scheint mir nach Abschluss der Maturaarbeit, ziemlich unmöglich zu sein. Nichtdestotrotz muss weiter geforscht werden, um evtl. doch einmal der zugrundeliegenden Ordnung auf die Spur zu kommen, sofern diese auch existent ist.

Wenn man am Abend zum Himmel hinaufschaut, erahnt man nicht welche Bewegungsfiguren die Planeten zueinander vollführen, erst die Software bringt die faszinierenden Muster zu Tage.

12. Zusammenfassung

Die vorliegende Maturaarbeit versucht dem Mysterium einer höheren Ordnung der Planetenbewegungen zueinander auf die Spur zu kommen. Mittels erstellten Computerprogramm konnte gezeigt werden, dass faszinierende Figuren wie Blumen und Sterne entstehen, die einen glauben lassen, dass tatsächlich eine höhere Ordnung existiert.

Das Computerprogramm gestattet es die Planetenkombinationen gegeneinander in Beziehung zu setzen und die berechneten Raumgeraden bzw. Geradenmittelpunkte graphisch darzustellen. In vielen Fällen kann gezeigt werden, dass eine mehr oder weniger regelmässige Figur entsteht. Im Fall der Erde-Venus-Blume scheint diese ‚fast' perfekt zu sein.

Da wir zwei Planeten vergleichen, welche sich auf elliptischen Bahnen bewegen, muss zwangsläufig auch die Kombinationsbewegung dieser beiden Planeten zueinander eine Art harmonische Bewegung generieren, so dass nicht wissenschaftlich fundiert gesagt werden kann, dass dahinter eine höhere Ordnung steht.

Die Untersuchungen müssten breiter erfolgen und die Bewegungsfiguren müssten zueinander in Verbindung gesetzt werden. Je weiter man seine Gedanken gehen lässt, umso komplexer erscheint es, einer Gesetzmässigkeit auf die Spur zu kommen.

Auf jeden Fall konnte mit dieser Maturaarbeit gezeigt werden, dass faszinierende Figuren entstehen. Im Zuge der begrenzten Zeit, war es leider nicht möglich, einer höheren Ordnung auf die Spur zu kommen - sofern diese überhaupt existiert.

13. Quellenverzeichnis

YouTube, Erde und Venus mit der Sonne als Zentrum, Yukterez Foundation, Jahr 2010, https://www.y-outube.com/watch?v=p5Q-RhlfjTw

YouTube, The dance of the Earth and Venus around the Sun, Francesco Carpinteri, 2018, https://www.youtube.com/watch?v=r_DYZWpp95g&list=PLv3YmrFKDVPGGPE5-hG2m4EjdmcdziUtO

YouTube, Earth Venus Tango around the Sun, Aubrey Meyer, Jahr 2011, https://www.youtube.com/watch?v=4cgQNUhtmHM&list=PLv3YmrFKDVPGGPE5hG2m4EjdmcdziUtO&index=2

14. Literaturverzeichnis

editing, t. f.-o. (2019, May 27). *VSOP*. Retrieved from Wiskipedia (Englische Version): https://en.wikipedia.org/wiki/VSOP_(planets)

Illustration of Kepler's laws of planetary motion with two planets. (2017, Oktober https://commons.wikimedia.org/wiki/File:Kepler_laws_diagram.svg#file). https://commons.wikimedia.org/wiki/File:Kepler_laws_diagram.svg#file, User:Rainald62.

Kepler, J. (1609, Heidelberg). *Astronomia nova ..., seu physica coelestis, tradita commentariis de motibus stellae martis.* Heidelberg: ETH-Bibliothek Zürich, Rar 4482, https://doi.org/10.3931/e-rara-558 / Public Domain Mark. Retrieved from Astronomia nova ..., seu physica coelestis, tradita commentariis de motibus stellae martis: Heidelberg : Voegelin, 1609. ETH-Bibliothek Zürich, Rar 4482, https://doi.org/10.3931/e-rara-558 / Public Domain Mark

Krafft, F. (2005). *Johannes Kepler Was die Welt im Innersten zusammenhält.* ISBN 3-86539-015-3: Marix Verlag.

Posch, T. (2017). *Johannes Kepler Die Entdeckung der Weltharmonie.* ISBN 978-3-8062-3452-7: Theiss Verlag.

Richter, D. (2017). *Ephemeridenrechnung Schritt für Schritt.* Berlin: Springer Verlag.

15. Abbildungsverzeichnis

Abbildung 1 - Universum Modell von Kepler 3
→ Bildnachweis: https://de.wikipedia.org/wiki/Johannes_Kepler [Stand: 24.11.2019]
Abbildung 2 - Portrait von Johannes Kepler im Alter von 39 Jahren 3
→ Bildnachweis: https://de.wikipedia.org/wiki/Johannes_Kepler [Stand: 24.11.2019]
Abbildung 3 - Planeten des Sonnensystems - nicht massstabstreue Darstellung der Planeten 5
→ Bildnachweis: https://de.wikipedia.org/wiki/Liste_der_Planeten_des_Sonnensystems [Stand: 24.11.2019]
Abbildung 4 - Darstellung der Keplerschen Gesetze 7
→ Bildnachweis: https://anthrowiki.at/Datei:Kepler_laws_diagram.svg [Stand: 25.11.2019]
Abbildung 5 - Merkur Fact Sheet 8
Abbildung 6 - Venus Fact Sheet 8
Abbildung 7 - Ellipsengeometrie Skizze 10
Abbildung 8 - Darstellung der Ellipsengeometrie 12
→ Bildnachweis: https://commons.wikimedia.org/w/index.php?curid=57546552 [Stand: 26.11.2019]
Abbildung 9 - Graphik der Erde-Venus Verbindungslinien über ein halbes Jahr 13
Abbildung 10 - Ephemeriden Daten der Venus von der NASA Internetseite 16
→ Bildnachweis: https://ssd.jpl.nasa.gov/horizons.cgi#top [20.11.2019]
Abbildung 11 - Programmablaufplan 17
Abbildung 12 - Quellcode des Vorprogrammes 18
Abbildung 13 - Erde-Venus Blume des Vorprogrammes 18
Abbildung 14 - Klasse des Programmes mit den Definitionen 19
Abbildung 15 - Import-Datenbanken 20
Abbildung 16 - Primärdaten 20
Abbildung 17 - Buttons des Konstruktors 21
Abbildung 18 - MenuBar des Konstruktors 22
Abbildung 19 - GPanel Definition 23
Abbildung 20 - "actionCallback" Methode Variablen 23
Abbildung 21 - "actionCallback" Methode Selektion 23
Abbildung 22 - "actionCallback" Methode Ellipsen-CheckBox 24
Abbildung 23 - "actionCallback" Methode Skalierung 24
Abbildung 24 - "actionCallback" Methode Graphik-Auswahl 25
Abbildung 25 - "printGPanel" Methode 25
Abbildung 26 - "clearGPanel" Methode 25
Abbildung 27 - "Legende" Methode 25
Abbildung 28 - "Winkel1" & "Winkel2" Methoden 26
Abbildung 29 - "Raumgerade" Methode 26
Abbildung 30 - "Mittelpunkt" Methode 27
Abbildung 31 - das Programminterface 28
Abbildung 32 - Venus-Erde Graphik 29
Abbildung 33 - Erde-Jupiter Graphik 30
Abbildung 34 - Jupiter-Saturn Graphik 31
Abbildung 35 - Mars-Jupiter Graphik 32
Abbildung 36 - Erde-Mars Graphik 33
Abbildung 37 - Saturn-Uranus Graphik 34
Abbildung 38 - Merkur-Venus Graphik 35

16. Tabellenverzeichnis

Tabelle 1; Planetendaten zum 3.Keplerschen Gesetz 7
Tabelle 2; Planetendaten der NASA sowie eigene Berechnungen grün hinterlegt........ 9
Tabelle 3; Orbitale Bahnneigungswinkel 11
Tabelle 4; Geschwindigkeitsverlauf ausgesuchter Planeten auf der Ellipsenbahn 12
Tabelle 5 ; Matrix der empirisch ermittelten Planetenkombinationseinstellungen 15

17. Nachwort und Danksagung

Bei dieser Arbeit wurde mir von mehreren Seiten geholfen.

So möchte ich mich an dieser Stelle bei Herrn Haas für seine Unterstützung in Programmierangelegenheiten bedanken. Durch seine Hinweise konnte die Funktionalität des Programms erweitert und bedienerfreundlicher gemacht werden.

Ebenso bedanke ich mich bei meinem Vater, der mich immer wieder ermutigt hat, tiefer in die Materie einzusteigen und nicht aufzugeben.

Überaus dankbar bin ich meinem Mentor Herrn Bernard für seine hilfreichen Hinweise und Gedankenanstösse während der Ausarbeitung der Maturaarbeit.

18. Anhang

18.1. Vorprogramm

```
from gpanel import *
from math import *

makeGPanel("ErdeVenusBlume",-150, 150, -150, 150)
#Erde
setColor("blue")
circle(149.60)
#Venus
setColor("green")
circle(108.21)

def RaumgeradeEV():
    t=974        #Anzahl 3-Tage Schritte (hier 8 Jahre)
    we=0.051606      #Winkel Erde in Bogenmass
    wv=0.083887      #Winkel Venus in Bogenmass
    for x in range(1,t,1):
        lineWidth(1)
        setColor("red")
        xErde = (math.cos(we*x))*149.60
        yErde = (math.sin(we*x))*149.60
        xVenus = (math.cos(wv*x))*108.21
        yVenus = (math.sin(wv*x))*108.21
        line(xErde, yErde, xVenus, yVenus)

RaumgeradeEV()
```

18.2. Programm

```
from math import *
from javax.swing import *
from gpanel import *

# P = Planeten; F = Farben; a = Ellipsenhalbachse a; b = Ellipsenhalbachse b; t = Umlaufzeit; g = GPanel-Koordinaten yMin, yMax, xMin, xMax
P = ["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus", "Neptune"]#Planeten: Mercury(0), Venus(1), Earth(2), Mars(3), Jupiter(4), Saturn(5), Uranus(6), Neptune(7)
F = ["Red", "Orange", "Yellow", "Lime", "Green", "Cyan", "Blue", "Magenta","Purple", "Black", "Gold"]#Farben: red(0), orange(1), yellow(2), lime(3), green(4), cyan(5), blue(6), magenta(7), purple(8), black(9)
a = [57.91, 108.21, 149.60, 227.92, 778.57, 1433.53, 2872.46, 4495.06]#(km*10^6)
b = [56.67, 108.21, 149.58, 226.92, 777.64, 1431.24, 2869.46, 4494.77]#(km*10^6)
t = [87.969, 224.701, 365.256, 686.980, 4332.589, 10759.22, 30685.4, 60189.0]#(Tage)
TagesSchritteMatrix = [[0,3,3,5,10,30,50],[3,0,3,5,10,10,15,50],[3,3,0,5,10,10,20,50],[3,5,5,0,10,10,50,50],[5,10,10,10,0,20,50,100],[10,10,10,10,20,0,50,300],[30,15,20,50,50,50,0,550],[50,50,50,50,100,300,550,0]]#(Tage)
AnzahlTagesSchritteMatrix = [[0,680,850,1150,867,1075,1030,1205],[680,0,974,2350,435,1080,2047,1205],[850,974,0,1120,438,1095,1535,1205],[1150,2350,1120,0,1305,1100,620,1205],[867,435,438,1305,0,1085,7350,607],[1075,1080,1095,1100,1085,0,4305,1000],[1030,2047,1535,620,7350,4305,0,2800],[1205,1205,1205,1205,607,1000,2800,0]]#(Tage)
g = 5000

class program:

    def printGPanel(self, event):
        printScreen(0.8)
        dispose()

    def clearGPanel(self,event):
        clear()

    def Legende(self, selectedIndex1, selectedIndex2):
        AnzahlJahresSchritte = AnzahlTagesSchritteMatrix[selectedIndex1][selectedIndex2]*TagesSchritteMatrix[selectedIndex1][selectedIndex2]/t[2]
        setColor("black")
        text(-4950, -4500, str(P[selectedIndex1] + ", " + P[selectedIndex2]))
        text(-4950, -4650, str(TagesSchritteMatrix[selectedIndex1][selectedIndex2]) + " Tages-Schritte")
        text(-4950, -4800, str(round(AnzahlJahresSchritte, 1)) + " Jahre Zeitspanne")

    def Winkel1(self, InputWinkel1, TagesSchritte):
        Winkel1def = 360/InputWinkel1*pi/180*TagesSchritte
        return Winkel1def

    def Winkel2(self, InputWinkel2, TagesSchritte):
        Winkel2def = 360/InputWinkel2*pi/180*TagesSchritte
```

```
        return Winkel2def

    def Raumgerade(self, AnzahlTagesSchritte, W1, W2, Planet1EllipsenseiteAneu, Planet1EllipsenseiteBneu, Planet2EllipsenseiteAneu, Planet2EllipsenseiteBneu):
        lineWidth(1)
        for x in range(1,AnzahlTagesSchritte,1):
            xLineP1 = (math.cos(W1*x))*Planet1EllipsenseiteAneu
            yLineP1 = (math.sin(W1*x))*Planet1EllipsenseiteBneu
            xLineP2 = (math.cos(W2*x))*Planet2EllipsenseiteAneu
            yLineP2 = (math.sin(W2*x))*Planet2EllipsenseiteBneu
            line(xLineP1, yLineP1, xLineP2, yLineP2)

    def Mittelpunkt(self, AnzahlTagesSchritte, W1, W2, Planet1EllipsenseiteAneu, Planet1EllipsenseiteBneu, Planet2EllipsenseiteAneu, Planet2EllipsenseiteBneu):
        lineWidth(3)
        for x in range(1,AnzahlTagesSchritte,1):
            xPoint = (((math.cos(W1*x))*Planet1EllipsenseiteAneu)+((math.cos(W2*x))*Planet2EllipsenseiteAneu))/2
            yPoint = (((math.sin(W1*x))*Planet1EllipsenseiteBneu)+((math.sin(W2*x))*Planet2EllipsenseiteBneu))/2
            point(xPoint, yPoint)

    def actionCallback(self,event):

        selectedIndex1 = self.ComboBox1.selectedIndex
        selectedIndex2 = self.ComboBox2.selectedIndex
        selectedC = self.ComboBoxFarbe.selectedIndex

        TagesSchritte = TagesSchritteMatrix[selectedIndex1][selectedIndex2]
        AnzahlTagesSchritte = AnzahlTagesSchritteMatrix[selectedIndex1][selectedIndex2]

        #Planet 1 Daten
        InputWinkel1 = t[selectedIndex1]
        W1 = self.Winkel1(InputWinkel1, TagesSchritte)
        Planet1EllipsenseiteA = a[selectedIndex1]
        Planet1EllipsenseiteB = b[selectedIndex1]

        #Planet 2 Daten
        InputWinkel2 = t[selectedIndex2]
        W2 = self.Winkel2(InputWinkel2, TagesSchritte)
        Planet2EllipsenseiteA = a[selectedIndex2]
        Planet2EllipsenseiteB = b[selectedIndex2]

        if Planet1EllipsenseiteA >= Planet2EllipsenseiteA:
            x = g/Planet1EllipsenseiteA
            Planet1EllipsenseiteAneu = Planet1EllipsenseiteA*x
            Planet1EllipsenseiteBneu = Planet1EllipsenseiteB*x
            Planet2EllipsenseiteAneu = Planet2EllipsenseiteA*x
            Planet2EllipsenseiteBneu = Planet2EllipsenseiteB*x
        else:
            x = g/Planet2EllipsenseiteA
```

```
            Planet1EllipsenseiteAneu = Planet1EllipsenseiteA*x
            Planet1EllipsenseiteBneu = Planet1EllipsenseiteB*x
            Planet2EllipsenseiteAneu = Planet2EllipsenseiteA*x
            Planet2EllipsenseiteBneu = Planet2EllipsenseiteB*x

        if selectedC == 0:
            setColor("red")
        elif selectedC == 1:
            setColor("orange")
        elif selectedC == 2:
            setColor("yellow")
        elif selectedC == 3:
            setColor("lime")
        elif selectedC == 4:
            setColor("green")
        elif selectedC == 5:
            setColor("cyan")
        elif selectedC == 6:
            setColor("blue")
        elif selectedC == 7:
            setColor("magenta")
        elif selectedC == 8:
            setColor("purple")
        elif selectedC == 9:
            setColor("black")
        elif selectedC == 10:
            setColor("gold")

        if self.CheckBoxE.isSelected():
            lineWidth(1)
            ellipse(Planet1EllipsenseiteAneu, Planet1EllipsenseiteBneu)
            ellipse(Planet2EllipsenseiteAneu, Planet2EllipsenseiteBneu)

        if self.RadioBtnR.isSelected():
            self.Raumgerade(AnzahlTagesSchritte, W1, W2, Planet1EllipsenseiteAneu, Planet1EllipsenseiteBneu, Planet2EllipsenseiteAneu, Planet2EllipsenseiteBneu)
        elif self.RadioBtnM.isSelected():
            self.Mittelpunkt(AnzahlTagesSchritte, W1, W2, Planet1EllipsenseiteAneu, Planet1EllipsenseiteBneu, Planet2EllipsenseiteAneu, Planet2EllipsenseiteBneu)
        else:
            print "Fehler"

        if self.CheckBoxL.isSelected():
            self.Legende(selectedIndex1, selectedIndex2)

        move(0, 0)

    def __init__(self):

        self.ComboBox1 = JComboBox(P)
        self.ComboBox2 = JComboBox(P)
```

```
        self.ComboBoxFarbe = JComboBox(F)

        self.CheckBoxE = JCheckBox("Ellipsen")
        self.CheckBoxL = JCheckBox("Legende")

        self.RadioBtnR = JRadioButton("Raumgerade")
        self.RadioBtnM = JRadioButton("Mittelpunkt")
        RadioButtonGroup = ButtonGroup()
        RadioButtonGroup.add(self.RadioBtnR)
        RadioButtonGroup.add(self.RadioBtnM)

        OkButton = JButton("OK", actionPerformed = self.actionCallback)
        ClearButton = JButton ("Clear", actionPerformed = self.clearGPanel)
        PrintButton = JButton ("Print", actionPerformed = self.printGPanel)

        MenuBar = JMenuBar()
        MenuBar.add(self.ComboBox1)
        MenuBar.add(self.ComboBox2)
        MenuBar.add(self.ComboBoxFarbe)
        MenuBar.add(self.CheckBoxE)
        MenuBar.add(self.CheckBoxL)
        MenuBar.add(self.RadioBtnR)
        MenuBar.add(self.RadioBtnM)
        MenuBar.add(OkButton)
        MenuBar.add(ClearButton)
        MenuBar.add(PrintButton)

        makeGPanel(MenuBar, -g, g, -g, g)

        title("Kepler's Sphaerengeometrie")
        resizable(False)
        windowSize(1000, 1000)
        windowPosition(10, 10)

        validate()

program()
```

18.3. NASA Fact Sheets

i. Fact Sheet ‚Merkur‘ der NASA
ii. Fact Sheet ‚Venus‘ der NASA
iii. Fact Sheet ‚Erde‘ der NASA
iv. Fact Sheet ‚Mars‘ der NASA
v. Fact Sheet ‚Jupiter‘ der NASA
vi. Fact Sheet ‚Saturn‘ der NASA
vii. Fact Sheet ‚Uranus‘ der NASA
viii. Fact Sheet ‚Neptun‘ der NASA

Abrufbar unter dem Link; https://nssdc.gsfc.nasa.gov/planetary/factsheet/index.html [Stand: 26.11.2019]

19. Selbstständigkeitserklärung

Mit meiner Unterschrift bestätige ich, Michael Voemel, dass die Arbeit selbstständig verfasst und in schriftliche Form gebracht worden ist, dass sich die Mitwirkung anderer Personen auf Beratung und Korrekturlesen beschränkt hat und dass alle verwendeten Unterlagen und Gewährspersonen aufgeführt sind. Ich gebe das Einverständnis, dass diese Arbeit zur Abklärung künftiger möglicher Plagiate Dritter beigezogen werden darf. Ich weiss, dass die erstellte Arbeit erst nach Abschluss der Maturaarbeit veröffentlicht und frühestens nach Ablauf der Rekursfrist verwertet werden darf.

Heerbrugg, den 29. November 2019

____

Michael Voemel

20. Kurzzusammenfassung

Die vorliegende Maturaarbeit von Michael Voemel (Klasse 4Pb, Kantonschule Heerbrugg, Betreuer Herr Bernard Carlo) befasst sich mit dem Thema „Kepler's Sphärengeometrie" und versucht dem Mysterium einer höheren Ordnung der Planetenbewegungen zueinander auf die Spur zu kommen. Mittels erstellten Computerprogramms konnte gezeigt werden, dass faszinierende Figuren, wie Blumen und Sterne, entstehen. Im Fall der Erde-Venus-Blume scheint diese ‚fast' perfekt zu sein. Da wir zwei Planeten vergleichen, welche sich auf elliptischen Bahnen bewegen, muss zwangsläufig auch die Kombinationsbewegung dieser beiden Planeten zueinander eine Art harmonische Bewegung generieren.