Lösungen zu Übungszettel

Zettel 1
1) $f(x) = e^x,\ g(x) = a * x$, $(e^{ax})' = e^{ax} a * 1$
2) $123456x^123455 + 21x^2 + 4x + 0 \qquad -\frac{x^{-4}}{3} \qquad e^{x^2+5}2x \qquad 2(3-x)^1 * (-1)$
3) $0 + 1 + 2x/2 + 3x^2/6.... = \sum_{i=1}^{\infty} \frac{x^{i-1}}{i!} i = \sum_{i=1}^{\infty} \frac{x^{i-1}}{(i-1)!} \overset{j=i-1}{=} \sum_{j=0}^{\infty} \frac{x^j}{j!}$ 4)

- Sitzt Neil Armstrong in der Rakete?
- Ist es ein senkrechter Start?
- t und x vermischt
- Falle: km/h und m/s
- Keine Rakete kann mit 20m/s losstarten
- Unklare Definition der Werte

$h'(0) = 20m/s$ also ja, da $20 * 3.6 > 60$
5) c, b

Zettel 2
1a) $\underbrace{-u'(t)}_{Zerfallsrate} = C * u(t)$, Lösung $u(t) = e^{-Ct}$
1b) $u'(t) = C * u(t)$, Rate $C = 1/h$ wenn t in Stunden (für ein vorhandenes kommt eins dazu)
Lösung: $u(t) = e^t > 2^t \implies$ Bakterien sind keine Bankzinsen
1c) $u'(t) = \underbrace{-u(t)}_{Bananen} \underbrace{-\frac{1}{u(t)}}_{Zombies}$
2) $v'(t) = h''(t) = -g$, $v'(t) = h''(t) = -g + Cv(t)^2$, $v'(t) = h''(t) = \frac{C}{h(t)^2}$ 3) siehe oben, ..., $-t^2 g$

Zettel 3
Hinweis: $u^0(t)$ hier ist 0 der Index, keine Potenz!
Berechnete Näherungen für $u_{Schritte} \approx u(1) = 2.718281828$
1b) Euler Verfahren $u_{k+1} = u_k + \Delta t * u_k$
Iter=1 $\implies \Delta t = 1$, $u_1 = 2$
Iter=2 $u_2 = 2.25$

Iter=4 $u_4 = \frac{625}{256} = 2.4414$
1a) Picard Iteration $u^{k+1} = u_0 + \int_0^t u^k(s)ds//\ u^0(t) = 1,\ u^0(1) = 1$
$u^1(t) = 1 + x,\ u^1(1) = 2$
$u^2(t) = 1 + x + x^2/2,\ u^2(1) = 2.5$
$u^3(t) = 1 + x + x^2/2 + x^3/6,\ u^3(1) = 2.66666$
$u^4(t) = 1 + x + x^2/2 + x^3/6 + x^4/24,\ u^4(1) = 2.7083$
1c) RK4 (Rechnet mehrere Näherungen für u übers Interval aus und berechnet damit das Intergral über das Interval genauer.)
$\tilde{u}_1 = 1$
$\tilde{u}_2 = 1 + 1/2 * 1 = 1.5$
$\tilde{u}_3 = 1 + 1/2 * 1.5 = 1.75$
$\tilde{u}_4 = 1 + 1 * 1.75 = 2.75$
$$u_1 = 1 + \underbrace{\frac{1}{6}(1 + 3 + 3.5 + 2.75) * 1}_{BesseresIntegral} = 2.708333$$

2)

Fehler	Euler	Picard	RK4
1	0.718	0.718	0.01
2	0.468	0.2818	
3		0.0516	
4	0.277	0.01	

Verdoppeln der Schritte halbiert beim Euler-Verfahren den Fehler.
Euler mit 1 Schritt stimmt mit 1. Picard überein.
4. Picard ähnlich gut (nicht gleich) wie RK4 $\rightarrow$ kein Zufall.

RK4 ist hat ähnlich hohen Aufwand wie 4 Euler Iterationen, ist aber besser.
Picard Iteration benötigt exaktes Integrieren, was meist nicht möglich ist (z.B. 10^9 dimensionale Probleme).
Info: RK4 hat bei doppelter Anzahl an Schritten 1/16 des vorigen Fehlers.

3) Fehlerverstärkung Differenzieren:
$$\frac{|\delta n \sin(nt)|}{|\delta \cos(nt)|} = n\frac{|\sin(nt)|}{|\cos(nt)|}\ ^{\max}\ \overset{beiderseits}{=}\ n\frac{1}{1}$$

Fehlerverstärkung Integrieren:

$$\frac{|\delta \sin(nt)/n|}{|\delta \cos(nt)|} = \frac{|\sin(nt)|}{n|\cos(nt)|} \text{ max } \overset{beiderseits}{=} \frac{1}{n}$$

Integrieren stabil, Differenzieren instabil → krasser Gegensatz zum händischen rechnen. Wenn man eine Funktion mit einem Rauschfehler differenzieren will, dann kommt Mist raus.
Für kleine n liegt das Maximum von $\sin(nt)$ nicht mehr im Interval, sodass Integrieren stabil bleibt.

Zettel 4

1) Mögliche Lösung $u(\phi) = \sin(\phi)$

$$x(\phi) = \frac{1}{\sin(\phi)} * cos(\phi) = \tan(\phi)$$
$$y(\phi) = \frac{1}{\sin(\phi)} * sin(\phi) = 1$$

Das ist eine Gerade (wer hätte das gedacht!)

2) Für $u'(0) = 0 = r'(0)$ und $u(0) = \frac{2}{3R} \implies r(0) = \frac{3R}{2}$ ist $u''(0) = 0$. Das heißt alles Ableitungen sind für diesen Lichtstrahl 0 uns somit ändert sich nichts. Das bedeutet, dass in einer Sphäre mit Radius $\frac{3R}{2}$ Lichtstrahlen beliebig oft um das schwarze Loch kreisen.
Im klassischen Fall wäre nur $u(0) = u'(0) = 0$ stationär, aber das ist nicht sinnvoll, da $r = 1/u$.

3)Wir suchen ein ϕ, sodass $u(\phi) = 1/r_{Kugel}$ gilt. In der Praxis hat sich ein Runge Kutta 4 Verfahren mit konstanter (durch Tests ermittelte) Schritweite bewährt. Hierbei rechnet man mit dem vektorwertigen System

$$v'(\phi) = -u(\phi) + \frac{3R}{2}(u(\phi))^2,$$
$$u'(\phi) = v(\phi).$$

Damit rechnet man soweit, bis die Iteration den Radius der Kugel überschreitet, also

$$u_k > \frac{1}{r_{Kugel}} > u_{k+1}$$

Wir wissen nun, dass sich der Schnittpunkt in diesem Interval befindet, aber wie bestimmt man ihn genau? Hier ein paar Vorschläge, wobei ϕ_k der Stützpunkt für u_k ist, also $u_k \approx u(\phi_k)$.

- Das zum u_k gehörige $\phi_k \implies$ sehr große Fehler
- Lineare Interpolation $\phi_k + \Delta\phi \frac{u_{Kugel} - u_k}{u_{k+1} - u_k}$. Entspricht dem Schnittpunkt einer Linie zwischen den Funktionswerten und der Konstanten auf dem Funktionsgraphen. Noch immer viel größerer Fehler als durch die RK4 Iterationen bis zum u_k. Mit vernünftigen Schrittweiten sind noch immer Fehler deutlich sichtbar (siehe Folien). Auch instabil bei wiederholter Anwendung.
- Möglicherweise Polynom-Interpolation höherer Ordnung unter Miteinbeziehung von $u_k, u_{k+1}, u'_k, u'_{k+1}$. Nicht ausgetestet, Nullstellen finden von Polynomen höherer Ordnung nicht direkt möglich.
- Newton-Verfahren zum Nullstellen $(\phi_k + \theta)$ finden.

$$\begin{aligned} \theta_0 &= 0 \\ \theta_{l+1} &= \theta_l + \frac{u(\phi_k + \theta_l)}{u'(\phi_k + \theta_l)} \end{aligned}$$

  Die Werte für $u(\phi_k + \theta_l), u'(\phi_k + \theta_l)$ können aus einem zusätzlichen RK4-Schritt mit Schrittweite θ_l ausgehend von ϕ_k gewonnen werden. Dieses Verfahren konvergiert sehr schnell (< 5 Iterationen) und sorgt für sehr kleine Fehler. Gegebenenfalls kann man auch vom rechten Rand starten ($\theta_0 = \Delta\phi$).

Dieses Verfahren kann auf einem normalen Laptop 2000 Lichtstrahlen mit einem Fehler von 10^{-8} in 2 Millisekunden berechnen. Damit man 60 Bilder pro Sekunde rechnen kann, hat man nur 16ms pro Bild Zeit, also sind diese Optimierungen auch wichtig.