

Numerische Methoden

Woche 3

David Nadlinger

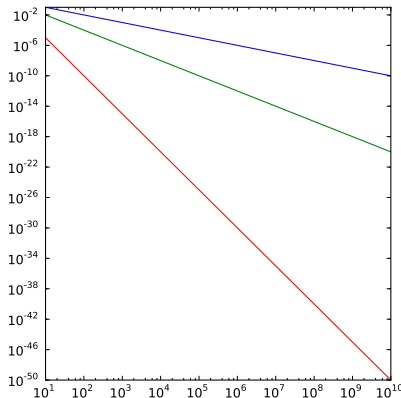
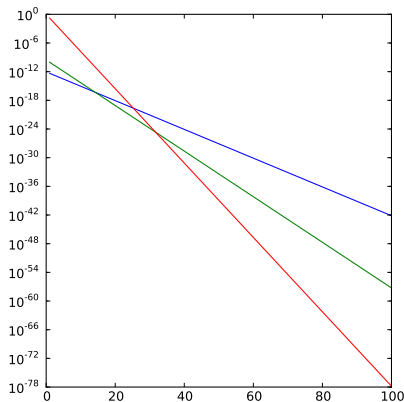
nadavid@ethz.ch

6. März 2013

Allgemeines

- Vorlesung? Serie 2? Python?
- Serien bitte drucken (aber gerne: 2 auf 1, *longbind*)
- Zusammenarbeiten: Ja, aber sinnvoll!
- <http://www.loria.fr/~rougier/teaching/matplotlib/>

Logarithmische Plots – was sind die Geraden?



Konvergenzordnung schätzen

$$p \approx \frac{\log \varepsilon_{k+1} - \log \varepsilon_k}{\log \varepsilon_k - \log \varepsilon_{k-1}}$$

Mögliche NumPy-Implementierung:

```
1 order_estimates =  
2     (np.log(es[2:]) - np.log(es[1:-1])) /  
3     (np.log(es[1:-1]) - np.log(es[:-2]))
```

Newton-Verfahren

Aufgabe 2

- Verwende generische Implementierung, z.B.:

```
1 from numpy.linalg import norm, solve
2
3 def newton(x0, F, DF, tol, maxit):
4     x = x0.copy()
5     for i in xrange(maxit):
6         s = solve(DF(x), F(x))
7         x -= s
8         if norm(s) < tol * (1 + norm(x)):
9             return x, i
10    raise Exception('Did not converge.')
```

- Erweitere um Rückgabe *aller* x -Werte
- Konvergenzrate schätzen wie besprochen

Aufgabe 3

- Verwende `matplotlib.pyplot.contour`:

```
1 x = np.linspace(..., num = n)
2 y = np.linspace(..., num = n)
3 [X, Y] = np.meshgrid(x, y)
4
5 vals = np.empty((n, n))
6
7 # Fülle vals[j, i] mit f(x[i], y[j]).
8
9 plt.contour(X, Y, vals, [0.0], colors='b')
10 plt.show()
```

Aufgabe 3 b

- Wähle x , suche y für $f(x, y) = 0$ mit Newton-Verfahren
- Abbruchkriterium unkritisch, z.B. Toleranz 10^{-4}
- Was sinnvoller Startwert wenn Funktion stetig?

Aufgabe 4

- Entweder: `plt.contourf` wie in Aufg. 3, siehe z.B. auch `np.mgrid`
- Oder: Erzeuge `np.zeros((N, N, 3))` für RGB-Farben, `plt.imshow` (jeder Kanal zwischen 0 und 1)
- Bonus: Verändere Helligkeit, etc. mit Iterationsanzahl bis Konvergenz
- Bonus: Andere Funktionen probieren, z.B. $z^5 = 1, \dots$