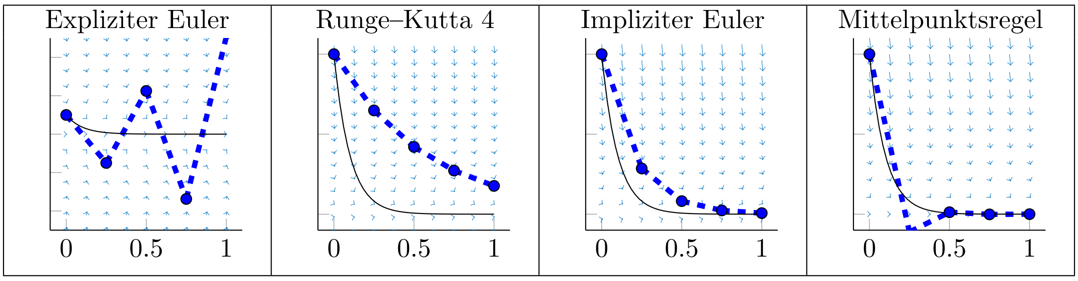
# Steife Probleme

Die Verfahren zur Lösung von ODEs lassen sich in zwei Kategorien unterteilen: explizite Lösungsverfahren und implizite Lösungsverfahren.
Explizite Lösungsverfahren lassen sich relativ einfach umsetzen und der mit der Lösung verbundene Rechenaufwand ist gering. So benötigt man bei dem expliziten Eulerverfahren eine Funktionsauswertung und bei dem Runge–Kutta–Verfahren vierter Ordnung vier Funktionsauswertungen pro Zeitschritt.
Implizite Lösungsverfahren benötigen im Gegensatz hierzu deutlich mehr Funktionsauswertungen, da hier eine Gleichung nach $x_{k+1}$ aufgelöst werden muss. Hierzu verwendet man zum Beispiel das Newtonverfahren, welches erst nach einigen Schritten eine Lösung liefert. Das bedeutet in jedem Zeitschritt müssen mehrere Schritte des Newtonverfahrens durchgeführt werden.
Dieser Mehraufwand ist dennoch unter bestimmten Voraussetzungen gerechtfertigt, und zwar, wenn das Problem welches man lösen möchte *steif* ist. Der Begriff der Steifigkeit ist ziemlich komplex. Um sich unter diesem Begriff dennoch etwas vorstellen zu können, folgt ein Beispiel.

Die folgende ODE ist für bestimmte Anfangswerte steif:

$$\dot{x} = -10x.$$

Für den Anfangswert $x_0 = 1$ lautet die analytische Lösung $x = e^{-10x}$. Wollen wir diese einfache ODE jedoch mit den bekannten Verfahren mit einer Schrittweite von $h = \frac{1}{4}$ lösen, so stellt uns das teilweise vor Probleme:




Die expliziten Verfahren "expliziter Euler" und "Runge–Kutta 4" liefern beide schlechtere Ergebnisse als die impliziten Verfahren "impliziter Euler" und "Mittelpunktsregel". Das liegt daran, dass implizite Verfahren grundsätzlich besser mit steifen Problemen umgehen können und dort auch keine Probleme mit der *Stabilität* der Lösung bekommen.
Wollen wir etwa die ODE mit dem expliziten Eulerverfahren lösen, so oszilliert die numerische Lösung um die x–Achse. Gleichzeitig wird die Entfernung der Lösung von der x–Achse mit jedem Schritt immer größer, obwohl das Richtungsfeld eindeutig die Lösung eigentlich in Richtung der x–Achse lenken sollte. Dieser Effekt tritt auf, weil die Schrittweite so groß ist, dass mit jedem Schritt die Lösung über das Ziel hinausschießt und aufgrund der großen Steigung auf der gegenüberliegenden Seite der x–Achse landet, wo unter Umständen (so wie hier) die Steigung sogar noch größer ist, was zu einer stetigen Vergrößerung des Fehlers führt. Verblüffenderweise liefert das implizite Eulerverfahren für die selbe Schrittweite deutlich bessere Werte. Auch das Runge–Kutta–Verfahren liefert, gemessen an der hohen Ordnung des Verfahrens, ein schlechtes Ergebnis, auch wenn keine Oszillationen, vergleichbar mit denen des expliziten Eulerverfahrens, zu beobachten sind. Dieser Vorteil von impliziten Verfahren rechtfertigt den höheren Rechenaufwand. Man müsste, um die Stabilitätsprobleme der expliziten Verfahren zu kompensieren, je nach Steifigkeit des Problems, extrem kleine Zeitschritte machen. Das würde wiederum viele Funktionsauswertungen bedeuten, die bei Verwendung des impliziten Verfahrens nicht notwendig sind.
In MATLAB kann man mit den Lösern `ode23` und `ode23s` ODEs innerhalb einer vorgegebenen Fehlertoleranz lösen. Löst man die ODE $\dot{x} = -\lambda x$ für unterschiedliche Werte von $\lambda$ wird deutlich, welche Bedeutung implizite Löser haben:

| $\lambda$ | $10^1$ | $10^2$ | $10^3$ | $10^4$ | $10^5$ |
|---|---|---|---|---|---|
| Anzahl Schritte mit `ode23` | 30 | 66 | 425 | 4006 | 39824 |
| Anzahl Schritte mit `ode23s` | 38 | 52 | 55 | 57 | 58 |