

Henon - eine alternative numerische Methode zur Ereignislokalisierung

Franz Preyser¹, Bernhard Heinzl², Felix Breitenecker¹

¹Institut für Analysis und Scientific Computing, TU Wien

²Institut für Rechnergestützte Automation, TU Wien

Franz.Preyser@tuwien.ac.at

In den folgenden Zeilen wird das Funktionsprinzip einer nicht-iterativen Methode zur Lokalisierung von Ereignissen, während des numerischen Lösens eines ODE-Systems, beschrieben. Weiters werden Ergebnisse eines Vergleichs, hinsichtlich Laufzeit, einer Matlab-Implementierung mit dem sonst üblichen Regula-Falsi-Verfahren präsentiert.

1 Einleitung

Beim numerischen Lösen von gewöhnlichen Differentialgleichungssystemen liefert das numerische Lösungsverfahren (ODE-Solver) Lösungswerte zu konkreten Werten der unabhängigen Variablen (meist die Zeit). Diese Werte haben entweder festen Abstand zueinander, oder die Abstände sind variabel (fix-step-/variable-step-solver). In manchen Fällen ist es jedoch notwendig, Lösungswerte bei ganz bestimmten Systemzuständen zu berechnen, um z.B. anschließend Parameterwerte oder Gleichungen zu ändern.

Die übliche Herangehensweise ist, eine sogenannte Ereignisfunktion $S: \Omega \rightarrow \mathbb{R}$ zu definieren. Diese Funktion bildet den Zustandsraum Ω stetig in die reellen Zahlen ab und ihre Nulldurchgänge kennzeichnen die Ereigniszeitpunkte. Das numerische Lösungsverfahren überprüft nun nach jeder Lösungswertberechnung, ob sich das Vorzeichen der Funktion S geändert hat. Wird ein Vorzeichenwechsel detektiert, so wird ein Algorithmus zur Nullstellenlokalisierung angeworfen. Iterative Verfahren sind das Newton-, das Sekanten-, das Bisektions- oder das Regula-Falsi-Verfahren. M. Henon stellt in [1] ein alternatives Verfahren vor, das über einen Wechsel der Integrationsvariablen versucht ohne Iterationen auszukommen.

2 Henon-Verfahren

Das Henon-Verfahren kann als eine Erweiterung des Newton-Verfahrens gesehen werden und stellt deshalb auch dieselben Ansprüche an die Funktion,

deren Nullstelle lokalisiert werden soll - hier die Ereignisfunktion. Vernachlässigt man den numerischen Fehler und geht also davon aus, dass die berechneten Lösungswerte x_n gleich den Werten der analytischen Lösung $x(t_n)$ sind, so kann die Ereignisfunktion auch als Funktion der Zeit gesehen werden: $\tau(t) = S(x(t))$. Ist nun τ in dem Intervall $(t_n, t_e + (t_e - t_n))$ stetig und monoton, wobei $\text{sign}(\tau(t_n)) \neq \text{sign}(\tau(t_{n+1}))$ und $\tau(t_e) = 0$ gelte und ist zusätzlich die Ableitung $d\tau/dt$ in dem Intervall bekannt, so darf Newton angewandt werden:

$$t_T = t_n - \left(\frac{d\tau}{dt}(t_n) \right)^{-1} \cdot \tau(t_n) \quad (1)$$

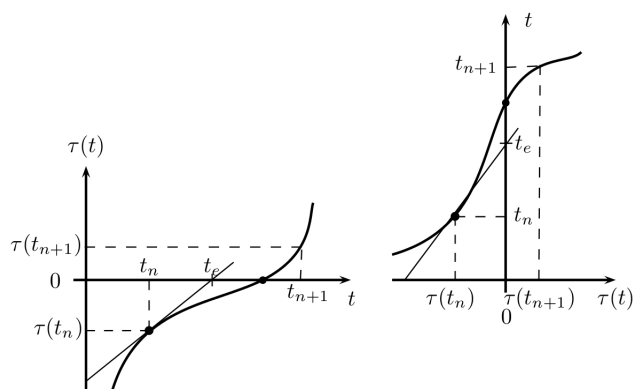


Abbildung 1. Eine Iteration im Newton-Verfahren bzw. im Henon-Verfahren unter Verwendung des expliziten Euler. li.: t als Integrationsvariable; re.: τ als Integrationsvariable

Gleichung (1) kann jedoch auch anders interpretiert werden. Da mit den obigen Voraussetzungen $\tau(t)$ im Intervall $(t_n, t_e + (t_e - t_n))$ invertierbar ist und die Inverse

im Punkt $\tau(t_n)$ die Steigung $(d\tau/dt(t_n))^{-1}$ hat, entspricht (1) genau einem Schritt des expliziten Eulerverfahrens mit Schrittweite $-\tau(t_n)$, angewandt auf das AWP:

$$\frac{dt}{d\tau}(\tau) = \left(\frac{d\tau}{dt}(t) \right)^{-1}, \quad t(\tau_n) = t_n \quad (2)$$

Es liegt daher nahe, anstelle der iterativen Anwendung des Newton-/Expliziten Euler-Verfahrens, einmalig ein Verfahren höherer Ordnung anzuwenden. Beim Henon-Verfahren passiert genau das. Für die Dauer der Ereignislokalisierung nimmt τ den Platz der unabhängigen Variable ein, während die Zeit t zu einer Funktion von τ wird. Anschaulich spiegelt man um die 1. Mediane und integriert dann im Ereigniswertebereich von $\tau=\tau(n)$ bis $\tau=0$ (siehe Abb. 1). Eine wesentliche Voraussetzung für Newton wie auch für Henon ist also, im Unterschied zu allen anderen aufgezählten Methoden, dass die Ableitung der Ereignisfunktion analytisch bekannt und die Ereignisfunktion in einem kleinen Bereich um die zu lokalisierende Nullstelle invertierbar sein muss.

3 Implementierung in Matlab

Um die Brauchbarkeit der Henon-Methode zu überprüfen, haben wir sie in Matlab implementiert. Konkret wurde eine Wrapper-Funktion für Matlab ODE-Solver geschrieben. Diese kann somit völlig universell auf jedes Beispiel angewandt werden, wo üblicherweise ein Matlab-Solver zum Einsatz kommt und die Voraussetzungen für das Newton-Verfahren erfüllt sind. Also AWP's der Bauart:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(t, \mathbf{x}), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad (3)$$

Es ist möglich, AWP (2) und (3) so zusammenzufassen, dass lediglich der Wert einer Variablen K entscheidet, ob beim Lösen des resultierenden AWP's (4), gerade das eigentliche Problem (3) integriert, oder eine Nullstelle der Ereignisfunktion berechnet wird.

$$\begin{aligned} \frac{d\mathbf{y}}{d\tau} &= \mathbf{f}(t(\tau), \mathbf{y}) \cdot K(\tau, \mathbf{y}) \\ \mathbf{y}(\tau_0) &= \mathbf{y}_0 \\ \frac{dt}{d\tau}(\tau) &= K(\tau, \mathbf{y}), \quad t(\tau_0) = t_0 \end{aligned} \quad (4)$$

Ist die Funktion K die konstante 1-Funktion, so gilt $\tau=t$ und somit auch $\mathbf{y}(\tau)=\mathbf{x}(t)$. Setzt man K gleich $(d\tau/dt)^{-1}$ und die Anfangswerte entsprechend, so

wechselt die Rolle der unabhängigen Variablen τ von Zeit auf Ereignisfunktion. Man integriert (4) dann bis $\tau=0$ und erhält somit den Systemzustand $\mathbf{x}(t_e)=\mathbf{y}(0)$ zum Ereigniszeitpunkt t_e , sowie t_e selbst.

4 Vergleiche

Verglichen wurde die Henon-Implementierung mit dem Regula-Falsi-Verfahren. Um den Vergleich fair zu gestalten, wurde nicht der ausgereifte Matlab-Lokalisierungsmechanismus verwendet, sondern eine eigene Implementierung analog zur Henon-Implementierung verwendet. Am Beispiel des Bouncing Ball mit unterschiedlichen Ereignisdefinitionen, wurde die Performance hinsichtlich gemachtem Fehler und der Laufzeit verglichen. Lokalisiert wurden die Nullstellen der Ball-Flugbahn sowie deren Extremstellen.

Beispiel	Solver	t_{RF}/t_H	t_{ML}/t_H
BB	ode23	8.75	0.19
BB M	ode23	0.60	0.11
d BB M	ode23	1.06	0.10
BB	ode45	4.76	0.35
BB M	ode45	4.43	0.35
d BB M	ode45	2.98	0.25

Tabelle 1. Ergebnis Laufzeitvergleich

t_H ... Laufzeit der Henon-Implementierung
 t_{RF} ... Laufzeit der Regula-Falsi-Implementierung
 t_{ML} ... Laufzeit des Matlab-Mechanismus
 BB ... Bouncing Ball
 BB M ... Bouncing Ball mit Maximumlokalisierung
 d BB ... gedämpfter Bouncing Ball

5 Fazit

Henon hat sich im Vergleich zu Regula Falsi als mindestens ebenbürtig erwiesen. Vor allem bei verhältnismäßig großer Schrittweite der zugrundeliegenden Integrationsmethode lassen die Ergebnisse auf Laufzeitvorteile hoffen. Es bedarf allerdings sicherlich noch diverser Optimierungen um mit ausgereiften Algorithmen mithalten zu können.

6 Referenzen

1. M. Hénon: *On the numerical Computation of Poincaré Maps*, Physica D: Nonlinear Phenomena, 5(2), S. 412–414, 1982.