

Zaawansowane metody numeryczne
Komputerowa analiza zagadnień różniczkowych

7. Metody Rungego-Kutty II

P. F. Góra

<http://th-www.if.uj.edu.pl/zfs/gora/>

semestr letni 2007/08

Metody DIRK

Jeśli spodziewamy się problemów ze stabilnością, w szczególności jeśli rozwiązyjemy problemy sztywne, powinniśmy stosować metody niejawne. Niejawna s -punktowa metoda Rungego-Kutty

$$\begin{array}{c|cccccc} \alpha_1 & \beta_{11} & \beta_{12} & \beta_{13} & \cdots & \beta_{1s} \\ \alpha_2 & \beta_{21} & \beta_{22} & \beta_{23} & \cdots & \beta_{2s} \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ \alpha_s & \beta_{s1} & \beta_{s2} & \beta_{s3} & \cdots & \beta_{ss} \\ \hline & w_1 & w_2 & w_2 & \cdots & w_s \end{array} \quad (1)$$

zastosowana do rozwiązania m -wymiarowego układu równań pierwszego rzędu, wymaga w każdym kroku rozwiązania układu $s \times m$ równań algebraicznych, w ogólności nieliniowych, co może być **BARDZO** kosztowne.

Aby zmniejszyć ten koszt nie tracąc stabilności metod niejawnych, stosuje się metody DIRK (ang. *Diagonally Implicit Runge-Kutta*), zwane też metodami pół-niejawnymi (ang. *semi-implicit*). W metodach tych macierz B ma niezerowe elementy diagonalne i poddiagonalne — elementy ponaddiagonalne są zerowe:

$$\begin{array}{c|cccccc}
 \alpha_1 & \beta_{11} & 0 & 0 & \cdots & 0 \\
 \alpha_2 & \beta_{21} & \beta_{22} & 0 & \cdots & 0 \\
 \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\
 \alpha_s & \beta_{s1} & \beta_{s2} & \beta_{s3} & \cdots & \beta_{ss} \\
 \hline
 & w_1 & w_2 & w_2 & \cdots & w_s
 \end{array} \tag{2}$$

Dzięki temu, stosując s -krokową metodę do problemu m -wymiarowego, zamiast rozwiązywać *jeden* układ równań algebraicznych o wymiarze $s \times m$, należy rozwiązać s układów o wymiarze m — po jednym układzie na każdy krok. Jest to numerycznie prostsze.

Przykłady metod DIRK podane były na ćwiczeniach.

Adaptacyjne podwajanie/połowienie kroku

Dobre algorytmy numerycznego całkowania ODE powinny, w miarę możliwości, same ustalać krok, z jakim przechodzą zadany przedział. W tym celu algorytm musi znać **oszacowanie błędu** popełnionego w ciągu jednego kroku. Dla metod Rungego-Kutty najprostszym tego typu algorytmem jest algorytm **adaptacyjnego podwajania/połowienia kroku**. Przypuśćmy, że żądam, aby błąd na jeden krok nie przekraczał Δ_{\max} . Niech aktualny krok całkowania wynosi h . Przechodzę przedział $[x_n, x_n + 2h]$ dwa razy: raz dwoma krokami o długości h i raz krokiem o długości $2h$.

W ten sposób otrzymuję *dwa* oszacowania wartości y_{n+1} . Dla metody rzędu p spełniają one

$$y(x_n + 2h) = y_{n+1}^{(2)} + 2h^{p+1}\phi + O(h^{p+2}), \quad (3a)$$

$$y(x_n + 2h) = y_{n+1}^{(1)} + (2h)^{p+1}\phi + O(h^{p+2}). \quad (3b)$$

Różnica

$$\Delta = y_{n+1}^{(1)} - y_{n+1}^{(2)} \quad (4)$$

stanowi oszacowanie błędu. Zachodzą dwa przypadki:

1. $\|\Delta\| \leq \Delta_{\max}$: Wówczas przechodzę do punktu $x_{n+1} = x_n + 2h$, jako rozwiązanie przyjmuję $y(x_n + 2h) = y_{n+1}^{(2)}$, zwiększam krok $h \rightarrow 2h$ i próbuję przejść następny przedział z dwa razy większym krokiem.
2. $\|\Delta\| > \Delta_{\max}$: Wówczas cofam się do punktu x_n , zmniejszam krok $h \rightarrow h/2$ i ponawiam całą procedurę. W tym wypadku należy się zabezpieczyć przed zmniejszeniem się kroku poniżej pewnego h_{\min} .

Istotne może być *jaka* normę weźmiemy.

Dla klasycznej metody czteropunktowej wymaga to $4 + 2 \cdot 4 - 1 = 11$ obliczeń prawej strony.

Lokalna ekstrapolacja

Jeśli popełniany błąd nie przekracza maksymalnego błędu dopuszczalnego, można jeszcze poprawić rozwiązanie za pomocą tak zwanej *lokalnej ekstrapolacji*: Przyjmijmy, że lewe strony obu równań (3) są sobie równe. Wówczas eliminując $h^{p+1}\phi$ otrzymujemy

$$y(x_n + 2h) = y_n^{(2)} + \frac{\Delta}{2^p - 1} + O(h^{p+2}). \quad (5)$$

Może to poprawić numeryczne własności rozwiązania, ale tak naprawdę zysk na rzędzie metody jest pozorny.

Zagnieżdżone (*embedded*) metody Rungego-Kutty

Rozważmy dwie jawne s -krokowe metody Rungego-Kutty:

$$\mathbf{y}_{n+1}^{(1)} = \mathbf{y}_n + h \sum_{i=1}^s w_i^{(1)} \mathbf{k}_i + O(h^{p+1}), \quad (6a)$$

$$\mathbf{y}_{n+1}^{(2)} = \mathbf{y}_n + h \sum_{i=1}^s w_i^{(2)} \mathbf{k}_i + O(h^{p+2}), \quad (6b)$$

$$\mathbf{k}_i = \mathbf{f} \left(x_n + \alpha_i h, \mathbf{y}_n + h \sum_{j=1}^{i-1} \beta_{ij} \mathbf{k}_j \right) \quad (6c)$$

Metody te *różnią się jedynie wagami*, mają takie same punkty pośrednie, a więc taki sam zestaw wektorów $\{\mathbf{k}_i\}$, a ich rzędy różnią się o jeden. Parami takimi są na przykład para Fehlberga i para Dormanda-Prince'a podane na ćwiczeniach.

Obliczając *jeden* zestaw pochodnych w punktach pośrednich, mamy oszacowanie błędu:

$$\Delta = \mathbf{y}_{n+1}^{(1)} - \mathbf{y}_{n+1}^{(2)} = h \sum_{i=1}^s \left(w_i^{(1)} - w_i^{(2)} \right) \mathbf{k}_i, \quad \|\Delta\| \sim h^{p+1}. \quad (7)$$

Skoro $\|\Delta\| \sim h^{p+1}$, to $\Delta_{\max} \sim h_{\max}^{p+1}$. Mam więc oszacowanie

$$h_{\max} = h \left(\frac{\Delta_{\max}}{\|\Delta\|} \right)^{1/(p+1)}. \quad (8)$$

Jeśli $h > h_{\max}$, *zmniejszam* krok $h \rightarrow h_{\max}$ i powtarzam bieżący krok. Jeśli $h < h_{\max}$, *zwiększam* krok $h \rightarrow h_{\max}$ i z powiększonym krokiem próbuję iść dalej. Metody zagnieżdżone pozwalają zatem na bardziej precyzyjną kontrolę kroku od podwajania/połowienia, wymagają także mniej obliczeń (s obliczeń funkcji f).

Pewien wariant metody punktu środkowego

Przypuśćmy, iż rozwiązując pewien problem Cauchy'ego decydujemy się wypro-
wadzać wyniki z krokiem H , krok ten jednak jest za duży do obliczeń (powodo-
wałby zbyt duży błąd). Wykonujemy więc m małych kroków o długości

$$h = H/m. \tag{9}$$

Stosujemy metodę

$$\mathbf{z}_0 \equiv \mathbf{y}_n, \quad (10a)$$

$$\mathbf{z}_1 = \mathbf{z}_0 + hf(x_n, \mathbf{z}_0) \quad (\text{Euler}) \quad (10b)$$

dla $j = 2, 3, \dots, m$

$$\mathbf{z}_j = \mathbf{z}_{j-2} + 2hf(x_n + (j-1)h, \mathbf{z}_{j-1}) \quad (10c)$$

(pochodna w punkcie środkowym)

$$\mathbf{y}(x_n + H) \simeq \mathbf{y}_{n+1} = \frac{1}{2} (\mathbf{z}_m + \mathbf{z}_{m-1} + hf(x_n + H, \mathbf{z}_m)) \quad (10d)$$

Metoda ta wymaga $m + 1$ obliczeń prawej strony równania na kroku o długości H . Można pokazać, iż *błąd* tej metody zawiera *tylko parzyste* potęgi h :

$$y(x_n + H) - y_{n+1} = \sum_{j=1}^{\infty} a_j h^{2j}. \quad (11)$$

W szczególności, jeśli m jest parzyste i zastosujemy metodę (10) dwa razy, z m i $m/2$ kroków, dostaniemy

$$y(x_n + H) \simeq \frac{4y_m - y_{m/2}}{3} + O(h^5) \quad (12)$$

a więc jest to metoda rzędu czwartego. ($y_{m/2}$ oznacza końcowy wynik po zastosowaniu $m/2$ kroków o długości $2H/m$, nie zaś połowę kroków o długości H/m .) Wyrażenia (11) i (12) bardzo przypominają analogiczne wyrażenia dla całkowania metodami ekstrapolacji Richardsona i Romberga, a skoro tak, narzuca się zastosowanie jakiegoś „ekstrapolacyjnego” algorytmu całkowania ODE.

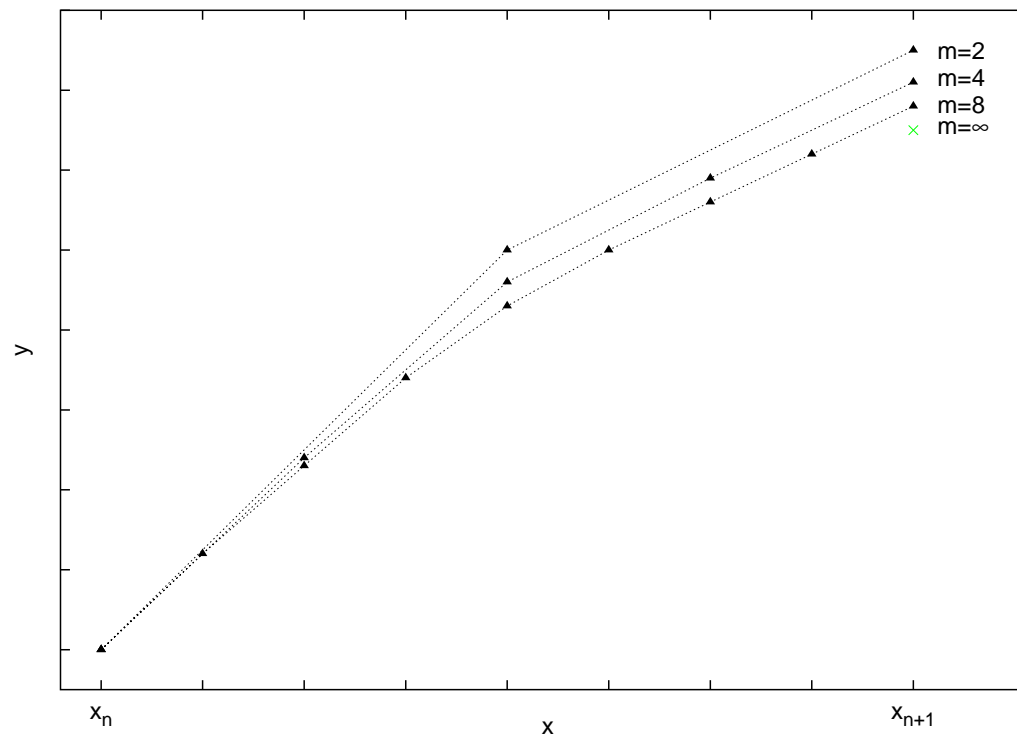
Metoda Bulirscha-Stoera

Przypuśmy, iż przechodzimy przedział $[x_n, x_n + H]$ przy użyciu metody (10) kolejno z krokami $h_m = H/m$ dla $m = 2, 4, 8, \dots$. Otrzymujemy w ten sposób *ciąg kolejnych przybliżeń* wartości $y(x_n + H)$. Jeśli tak otrzymany ciąg *ekstrapolujemy* do nieskończonej liczby kroków pośrednich — lub też, w innym sformułowaniu, *do* $h_m \rightarrow 0$ — oszacujemy jak powinno wyglądać numeryczne rozwiązanie badanego ODE w granicy infinitezymalnie małych kroków.

Możemy użyć ekstrapolacji wielomianowej lub poprzez funkcje wymierne. Cała ta procedura jest dość (ale nie przesadnie) kosztowna i warto ją stosować, gdy

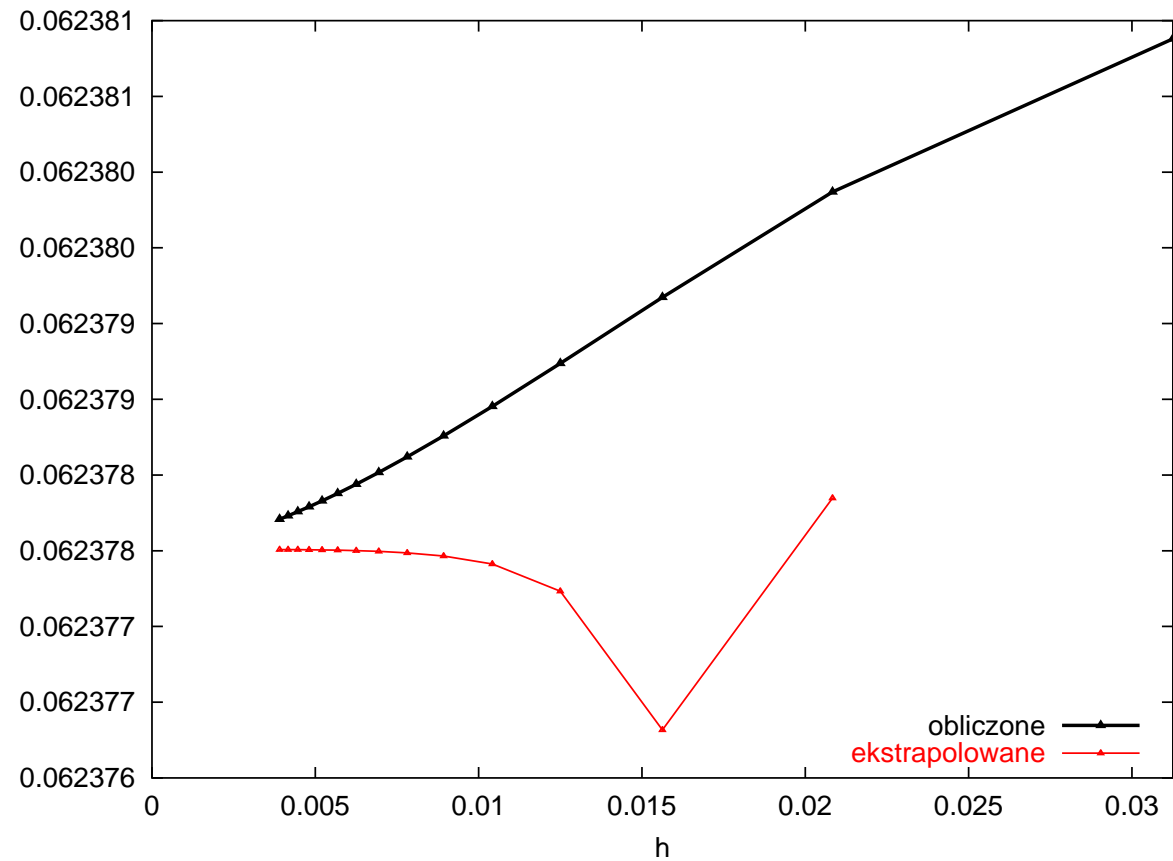
- zależy nam na szczególnie dokładnych rozwiązaniach,
- prawa strona równania zmienia się powoli.

Podobnie jak w ekstrapolacji Richardsona, w zasadzie podejście to można stosować tylko gdy ciąg kolejnych przybliżeń jest monotoniczny. Proces dzielenia odcinka kończymy gdy błąd ekstrapolacji jest mniejszy od zadanej tolerancji.

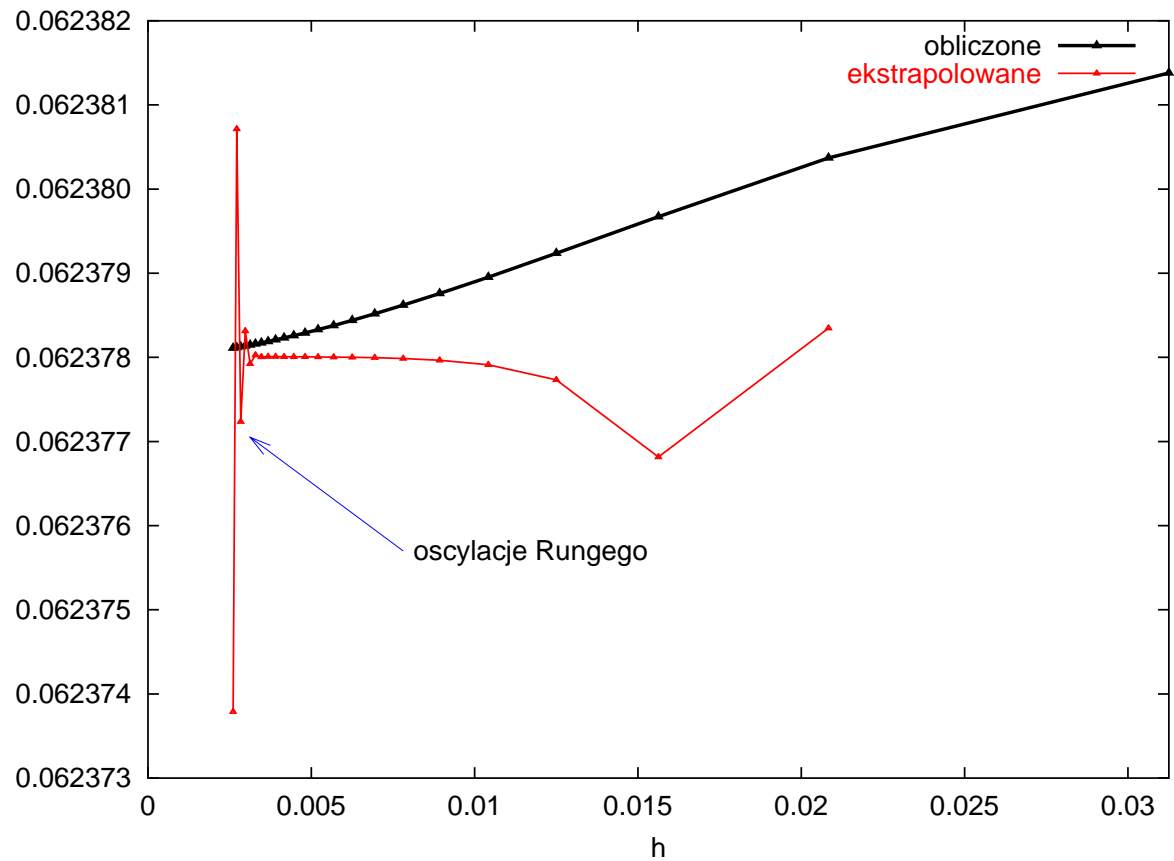


$\log_2 m$	obliczenie	ekstrapolacja
2	0.062381380134158664	
3	0.062380369335862672	0.062378347739270680
4	0.062379672556098376	0.062376816694340424
5	0.062379237240341216	0.062377733233895008
6	0.062378953354256184	0.062377912906545856
7	0.062378758903003136	0.062377965285744048
8	0.062378619973785528	0.062377985927438168
9	0.062378517198083888	0.062377995625316072
10	0.062378438952554864	0.062378000712959168
11	0.062378377939856608	0.062378003602874432
12	0.062378329395298192	0.062378005345019248
13	0.062378290100963920	0.062378006454798296
14	0.062378257819309512	0.062378007179444864
15	0.062378230954631280	0.062378007643411944
16	0.062378208343525576	0.062378007961218176

Graficzna ilustracja powyższych danych:

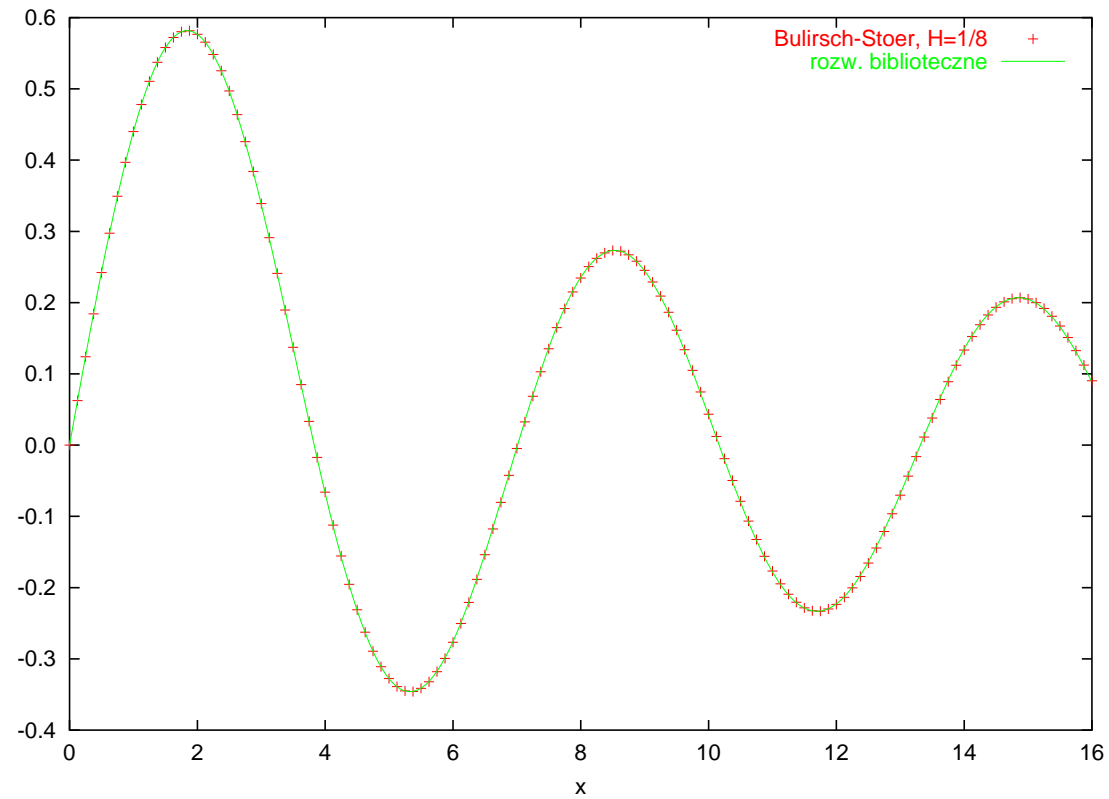


Zbyt duży stopień wielomianu ekstrapolacyjnego:



Ekstrapolacja funkcjami wymiernymi byłaby lepsza.

Przykład zastosowania metody Bulirsch-Stoera:



$$x^2 \frac{d^2 y}{dx^2} + x \frac{dy}{dx} + (x^2 - 1)y = 0$$

Odpowiednik metody Bulirscha-Stoera dla układów sztywnych

Jeśli spodziewamy się problemów ze stabilnością, używamy metod niejawnych, czasami jednak dokonujemy *pseudolinearyzacji* celem uproszczenia obliczeń. Punktem wyjścia jest następująca metoda niejawna:

$$\begin{aligned} \mathbf{y}_{n+1} - \mathbf{y}_{n-1} &= 2h\mathbf{f}\left(x_n, \frac{\mathbf{y}_{n+1} + \mathbf{y}_{n-1}}{2}\right) \\ &= 2h\mathbf{f}\left(x_n, \mathbf{y}_n + \underbrace{\frac{\mathbf{y}_{n+1} + \mathbf{y}_{n-1}}{2} - \mathbf{y}_n}_{\text{poprawka}}\right). \end{aligned} \quad (13)$$

Rozwijam prawą stronę (13) wokół (x_n, y_n) i porządkuję wyrazy:

$$\begin{aligned} \left[\mathbb{I} - h \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \Big|_{x_n, y_n} \right] \mathbf{y}_{n+1} &= \left[\mathbb{I} + h \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \Big|_{x_n, y_n} \right] \mathbf{y}_{n-1} \\ &+ 2h \left[\mathbf{f}(x_n, y_n) - h \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \Big|_{x_n, y_n} \mathbf{y}_n \right]. \end{aligned} \quad (14)$$

Powyższe wyrażenie jest *liniowym* (w odróżnieniu od nieliniowego wyrażenia (13)) równaniem na nieznaną wielkość \mathbf{y}_{n+1} . Opierając się na (14), można skonstruować odpowiednik metody (10). Będą w nim występowały Jakobiany $\partial \mathbf{f} / \partial \mathbf{y}$ obliczane w kolejnych punktach pośrednich. **Dokonuję następnego uproszczenia: wszystkie Jakobiany wyliczam w lewym krańcu „dużego” przedziału.** Ostatecznie dostaję ($h = H/m$):

$$\mathbf{H} = \left[\mathbb{I} - h \left. \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \right|_{x_n, \mathbf{y}_n} \right], \quad (15a)$$

$$\mathbf{z}_0 = \mathbf{y}_n, \quad (15b)$$

$$\Delta_0 = \mathbf{H}^{-1} h \mathbf{f}(x_n, \mathbf{z}_0), \quad (15c)$$

$$\mathbf{z}_1 = \mathbf{z}_0 + \Delta_0, \quad (15d)$$

$$\Delta_j = \Delta_{j-1} + 2\mathbf{H}^{-1} \left[\mathbf{f}(x_n + jh, \mathbf{z}_j) - \Delta_{j-1} \right], \quad (15e)$$

$$\mathbf{z}_{j+1} = \mathbf{z}_j + \Delta_j, \quad (15f)$$

dla $j = 1, 2, \dots, m - 1$

$$\Delta_m = \mathbf{H}^{-1} \left[\mathbf{f}(x_n + H, \mathbf{z}_m) - \Delta_{m-1} \right], \quad (15g)$$

$$\mathbf{y}(x_n + H) \simeq \mathbf{z}_m + \Delta_m. \quad (15h)$$

Uwaga: Zapis postaci

$$\mathbf{q} = \mathbf{H}^{-1} \mathbf{p} \quad (16)$$

należy rozumieć symbolicznie — w praktyce w celu wyliczenia \mathbf{q} *należy* rozwiązać równanie

$$\mathbf{H}\mathbf{q} = \mathbf{p}. \quad (17)$$

Równanie (17) można rozwiązać na przykład metodą rozkładu LU . Ponieważ zgodnie z przyjętym uproszczeniem macierz \mathbf{H} jest taka sama we wszystkich krokach iteracji (15), rozkładu tego można dokonać *tylko raz*.

Dalej postępuję tak, jak poprzednio: Zagęszczam podział i stosuję ekstrapolację do $h \rightarrow 0$.