Matematyka obliczeniowa

dr inż. Piotr Piela

Wydział Informatyki ZUT w Szczecinie

Cele przedmiotu

Cele przedmiotu

- Ukształtowanie umiejętności dobierania właściwych algorytmów numerycznych w zależności od postawionego zadania.

Cele przedmiotu

- Ukształtowanie umiejętności dobierania właściwych algorytmów numerycznych w zależności od postawionego zadania.
- Ukształtowanie umiejętności zmniejszania wpływu błędu obliczeń numerycznych na wynik końcowy.

Cele przedmiotu

- Ukształtowanie umiejętności dobierania właściwych algorytmów numerycznych w zależności od postawionego zadania.
- Ukształtowanie umiejętności zmniejszania wpływu błędu obliczeń numerycznych na wynik końcowy.
- Ukształtowanie umiejętności tworzenia programów komputerowych wykorzystujących algorytmy numeryczne w różnego rodzaju zadaniach.

Efekty uczenia się

WIEDZA

- W wyniku przeprowadzonych zajęć student powinien być w stanie dobierać algorytmy obliczeniowe w zależności od postawionego zadania uwzględniając ich złożoność obliczeniową i szybkość działania.
- Student po zakonczonym kursie będzie potrafił wskazać miejsca generowania błędów w obliczeniach numerycznych i będzie potrafił zaproponować sposoby ograniczania tych błędów.

UMIEJĘTNOŚCI

- W wyniku przeprowadzonych zajęć student powinien umieć samodzielnie rozwiązywać problemy obliczeniowe w zakresie ograniczania wpływu błędów na wyniki, doboru algorytmów obliczeniowych a także do ich realizacji w wybranym środowisku programistycznym.

Literatura

- Kincaid D., Cheney W., *Analiza numeryczna*, WNT, Warszawa, 2006
- Kiełbasiński A., Schwetlick H., *Numeryczna algebra liniowa*, WNT, Warszawa, 1992
- Fortuna Z., Macukow B., Wąsowski J., *Metody numeryczne*, WNT, Warszawa, 1993
- Bożek B., *Metody obliczeniowe i ich komputerowa realizacja*, Wydawnictwa AGH, Kraków, 2005
- Matulewski J., Dziubak T., Sylwestrzak M., Płoszajczak R., *Grafika, Fizyka, Metody numeryczne*, PWN, Warszawa, 2010
- Palczewski A., *Równania różniczkowe zwyczajne*, WNT, Warszawa, 2004
- Popov O., *Metody numeryczne i optymalizacja*, Wydawnictwo Uczelniane Politechniki Szczecińskiej, Szczecin, 2003

Obliczenia numeryczne

Numeryczna reprezentacja liczb

- Zapis stałopozycyjny
- Zapis zmiennopozycyjny

Obliczenia numeryczne

Numeryczna reprezentacja liczb

- Zapis stałopozycyjny
- Zapis zmiennopozycyjny

Arytmetyka komputerowa

- Działania arytmetyczne na liczbach zmiennopozycyjnych
- Stabilność i uwarunkowanie algorytmu
- Błędy obliczeń

Obliczenia numeryczne

Numeryczna reprezentacja liczb
- Zapis stałopozycyjny
- Zapis zmiennopozycyjny

Arytmetyka komputerowa
- Działania arytmetyczne na liczbach zmiennopozycyjnych
- Stabilność i uwarunkowanie algorytmu
- Błędy obliczeń

Problemy z obliczeniami
- Zaokrąglanie i ucinanie

Obliczenia numeryczne

Numeryczna reprezentacja liczb
- Zapis stałopozycyjny
- Zapis zmiennopozycyjny

Arytmetyka komputerowa
- Działania arytmetyczne na liczbach zmiennopozycyjnych
- Stabilność i uwarunkowanie algorytmu
- Błędy obliczeń

Problemy z obliczeniami
- Zaokrąglanie i ucinanie

Numeryczne rozwiązywanie zadań

Numeryczna reprezentacja liczb

Liczby są reprezentowane przez skończoną liczbę cyfr ich rozwinięć pozycyjnych.

Numeryczna reprezentacja liczb

Liczby są reprezentowane przez skończoną liczbę cyfr ich rozwinięć pozycyjnych.

Najczęściej stosowane systemy liczbowe		
System liczbowy	Podstawa	Ustalone cyfry
dwójkowy	2	0,1
ósemkowy	8	0,1,2,3,4,5,6,7
szesnastkowy	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
dziesiętny	10	0,1,2,3,4,5,6,7,8,9

Numeryczna reprezentacja liczb

Liczby są reprezentowane przez skończoną liczbę cyfr ich rozwinięć pozycyjnych.

Najczęściej stosowane systemy liczbowe		
System liczbowy	Podstawa	Ustalone cyfry
dwójkowy	2	0,1
ósemkowy	8	0,1,2,3,4,5,6,7
szesnastkowy	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
dziesiętny	10	0,1,2,3,4,5,6,7,8,9

Przykład

$(123.756)_{10},\ (AF)_{16},\ (0.1011)_2$

Przy przejściu z jednego systemu liczbowego do innego stosuje się procedury konwersji.

Konwersja liczb

Konwersja liczby z układu dziesiętnego do układu dwójkowego.

$139 : 2$	$=$	69	reszta 1
$69 : 2$	$=$	34	reszta 1
$34 : 2$	$=$	17	reszta 0
$17 : 2$	$=$	8	reszta 1
$8 : 2$	$=$	4	reszta 0
$4 : 2$	$=$	2	reszta 0
$2 : 2$	$=$	1	reszta 0
$1 : 2$	$=$	0	reszta 1

Liczba $(139)_{10} = (10001011)_2$

Konwersja liczb

Konwersja liczby z układu dziesiętnego do układu dwójkowego.

$139 : 2$	$=$	69	reszta 1
$69 : 2$	$=$	34	reszta 1
$34 : 2$	$=$	17	reszta 0
$17 : 2$	$=$	8	reszta 1
$8 : 2$	$=$	4	reszta 0
$4 : 2$	$=$	2	reszta 0
$2 : 2$	$=$	1	reszta 0
$1 : 2$	$=$	0	reszta 1

Liczba $(139)_{10} = (10001011)_2$

Inne przykłady

$$(43)_{10} = (2B)_{16} = (53)_8 = (101011)_2$$
$$(25)_{10} = (19)_{16} = (31)_8 = (11001)_2$$

Konwersja liczb

Konwersja ułamka z układu dziesiętnego do układu dwójkowego.

$$\begin{array}{rcll} 0.8125 \cdot 2 & = & 1.625 & (1 = z_{-1}) \\ 0.625 \cdot 2 & = & 1.25 & (1 = z_{-2}) \\ 0.25 \cdot 2 & = & 0.5 & (0 = z_{-3}) \\ 0.5 \cdot 2 & = & 1.0 & (1 = z_{-4}) \\ 0 \cdot 2 & = & 0 & \end{array}$$

Liczba $(0.8125)_{10} = (0.1101)_2$

Konwersja liczb

Konwersja ułamka z układu dziesiętnego do układu dwójkowego.

$$\begin{array}{rcll} 0.8125 \cdot 2 & = & 1.625 & (1 = z_{-1}) \\ 0.625 \cdot 2 & = & 1.25 & (1 = z_{-2}) \\ 0.25 \cdot 2 & = & 0.5 & (0 = z_{-3}) \\ 0.5 \cdot 2 & = & 1.0 & (1 = z_{-4}) \\ 0 \cdot 2 & = & 0 & \end{array}$$

Liczba $(0.8125)_{10} = (0.1101)_2$

Inne przykłady

$(0.1)_{10} = (0.0001100110011001...)_2$
$(0.125)_{10} = (0.001)_2$

Zapis stałopozycyjny

Do zapisu liczby stałopozycyjnej przeznaczona jest z góry określona ilość cyfr dwójkowych (bitów), a pozycję przecinka ustala się arbitralnie, w zależności od wymaganej dokładności.

Dowolną liczbę całkowitą l możemy przedstawić w postaci rozwinięcia dwójkowego:

$$l = s \sum_{i=0}^{n} e_i \cdot 2^i \quad e_n \neq 0 \ dla \ l \neq 0,$$

gdzie:
s - znak liczby,
e_i - cyfry dwójkowe ($e_i = 0$ lub $e_i = 1$).

Jeśli na reprezentację liczby przeznaczymy $d + 1$ bitów to możemy **dokładnie** reprezentować liczby całkowite z przedziału:

$$\langle -2^d + 1, 2^d - 1 \rangle.$$

Zapis zmiennopozycyjny

Dowolną liczbę rzeczywistą $x \neq 0$ możemy przedstawić w postaci rozwinięcia dwójkowego:

$$x = s \cdot 2^c \cdot m,$$

gdzie:
s - znak liczby,
c - liczba całkowita - cecha,
m - liczba rzeczywista z przedziału $< 0.5, 1 >$ - mantysa,

Cechę zapisuje się w sposób stałopozycyjny na $d - t$ bitach słowa. Mantysę na pozostałych t bitach.

Znak	**Cecha**				**Mantysa**			
s	$e_{(d-t-1)}$	$e_{(d-t-2)}$	$\ldots$	e_0	e_{-1}	e_{-2}	$\ldots$	e_{-t}

Standard IEEE 754

Standard IEEE 754 określa normy dla arytmetyki implementowanej na komputerach, zawiera ustalenia dotyczące formatów, procedur zaokrąglania, operatorów arytmetycznych, konwersji liczb, postępowania w sytuacjach wyjątkowych, np. przy przekroczeniu zakresu liczbowego.

Standard IEEE 754 definiuje dwie klasy liczb:

- pojedynczej precyzji (*single*),
- podwójnej precyzji (*double*).

Parametr	**Pojedyncza precyzja**	**Podwójna precyzja**
Liczba bitów znaku	1	1
Liczba bitów cechy	8	11
Liczba bitów mantysy	23	52
Długość słowa w bitach	32	64

Standard IEEE 754 symbole specjalne

W standardzie IEEE 754 określono sposób zapisu symboli specjalnych:

Wartość specjalna	bit znaku	Bity cechy	Bity mantysy
NaN	x	111...111	xxx...xxx
$0+$	1	000...000	000...000
$0-$	0	000...000	000...000
$+\infty$	1	111...111	000...000
$-\infty$	0	111...111	000...000

Zapis zmiennopozycyjny

W idealnych rozważaniach mantysa wynosi:

$$m = \sum_{i=1}^{\infty} e_{-i} \cdot 2^{-i} \quad e_{-1} = 1,\ e_{-i} = 0 \ lub\ 1 \ dla\ i > 1,$$

Jednak w rzeczywistości:

$$m_t = \sum_{i=1}^{t} e_{-i} \cdot 2^{-i} + e_{-(t+1)} \cdot 2^{-t},$$

Popełniany błąd wynosi:

$$|m - m_t| \leqslant \frac{1}{2} \cdot 2^{-t}.$$

Liczby maszynowe

Precyzję arytmetyki dla danego typu komputera i stosowanej arytmetyki określa liczba:

$$\varepsilon = 2^{-t-1}.$$

gdzie:
t - liczba bitów mantysy.

W Matlabie precyzję arytmetyki ε określa zmienna predefiniowana *eps*

Liczba maszynowa $fl(x)$ jest to liczba rzeczywista x zapisana dokładnie w wybranej arytmetyce na skończonej liczbie bitów.

Nie każda liczba rzeczywista jest liczbą maszynową.

Liczby maszynowe

Precyzję arytmetyki dla danego typu komputera i stosowanej arytmetyki określa liczba:

$$\varepsilon = 2^{-t-1}.$$

gdzie:
t - liczba bitów mantysy.

W Matlabie precyzję arytmetyki ε określa zmienna predefiniowana *eps*

Liczba maszynowa $fl(x)$ jest to liczba rzeczywista x zapisana dokładnie w wybranej arytmetyce na skończonej liczbie bitów.

Nie każda liczba rzeczywista jest liczbą maszynową.

Przykłady

$(0.1)_{10} = (0.0001100110011001...)_2,\ \pi,\ e$

Elementy teorii błędu

Błąd bezwzględny - różnica pomiędzy dokładną rzeczywistą wartością x i jej wartością przybliżoną $\hat{x}$

$$\Delta_x = |x - \hat{x}|$$

Błąd względny (dla $x \neq 0$):

$$\delta_x = \frac{|x - \hat{x}|}{|x|}$$

Błąd względny w zapisie zmiennopozycyjnym wynosi:

$$\frac{|x - fl(x)|}{|x|} \leqslant \varepsilon.$$

Różnica między błędem względnym a bezwzględnym

Oblicz błąd względny i bezwzględny dla zmiennej x i y:

$$\begin{aligned} x &= 100, \quad \hat{x} = 100.1 \\ y &= 0.1, \quad \hat{y} = 0.2 \end{aligned}$$

Błąd bezwzględny:

$$\begin{aligned} \Delta_x &= |x - \hat{x}| = |100 - 100.1| = 0.1 \\ \Delta_y &= |y - \hat{y}| = |0.1 - 0.2| = 0.1 \end{aligned}$$

Błąd względny:

$$\begin{aligned} \delta_x &= \frac{|x-\hat{x}|}{|x|} = \frac{|100-100.1|}{|100|} = \frac{0.1}{100} = 0.001 \cdot 100\% = 0.1\% \\ \delta_y &= \frac{|y-\hat{y}|}{|y|} = \frac{|0.1-0.2|}{|0.1|} = \frac{0.1}{0.1} = 1 \cdot 100\% = 100\% \end{aligned}$$

Zapis zmiennopozycyjny - zakres

Liczba cyfr mantysy decyduje o dokładności przedstawienia liczb, liczba cyfr cechy określa zakres reprezentowanych liczb.

Jeśli założymy, że cecha:

$$c \in \langle c_{min}, c_{max} \rangle$$

gdzie:
$c_{max} = -c_{min} = 2^{d-t} - 1$ - liczba bitów mantysy

i mantysa:

$$m_t \in \langle \frac{1}{2}, 1 \rangle$$

to możemy reprezentować liczby x z zakresu:

$$\frac{1}{2} \cdot 2^{c_{min}} \leqslant |x| < 2^{c_{max}}.$$

Niedomiar i nadmiar

Niedomiar – działanie, którego wynik jest różny od zera, ale ma zbyt małą cechę.

W arytmetyce IEEE 754 liczby dla których następuje niedomiar są reprezentowane jako $0+$ lub $0-$

Niedomiar i nadmiar

Niedomiar – działanie, którego wynik jest różny od zera, ale ma zbyt małą cechę.

W arytmetyce IEEE 754 liczby dla których następuje niedomiar są reprezentowane jako $0+$ lub $0-$

Nadmiar – działanie, którego wynik ma zbyt dużą cechę.

W arytmetyce IEEE 754 liczby dla których następuje nadmiar są reprezentowane przez specjalną wartość Inf ($\pm\infty$), która propaguje się w obliczeniach zgodnie z powszechnie przyjętymi regułami, np: $x + \infty = \infty$, $x \cdot \infty = \infty$, $1/\infty = 0$, $\infty - \infty = NaN$, itd.

Jeśli nieskończoność jest argumentem działania, które nie ma sensu, to wykonanie programu jest automatycznie przerwane.

Działania arytmetyczne na liczbach zmiennopozycyjnych

W dobrze zaprojektowanym komputerze wyniki czterech działań na liczbach maszynowych spełniają związek:

$$fl(x \odot y) = (x \odot y)(1 + \delta), \quad |\delta| \leqslant \varepsilon$$

Symbol $\odot$ oznacza jedno z czterech podstawowych działań arytmetycznych

Działania arytmetyczne na liczbach zmiennopozycyjnych

W dobrze zaprojektowanym komputerze wyniki czterech działań na liczbach maszynowych spełniają związek:

$$fl(x \odot y) = (x \odot y)(1 + \delta), \quad |\delta| \leqslant \varepsilon$$

Symbol $\odot$ oznacza jedno z czterech podstawowych działań arytmetycznych

Jeśli x i y nie są liczbami maszynowymi to otrzymujemy:

$$fl(fl(x) \odot fl(y)) = (x(1 + \delta_1) \odot y(1 + \delta_2))(1 + \delta_3), \quad |\delta_i| \leqslant \varepsilon$$

Stabilność algorytmu

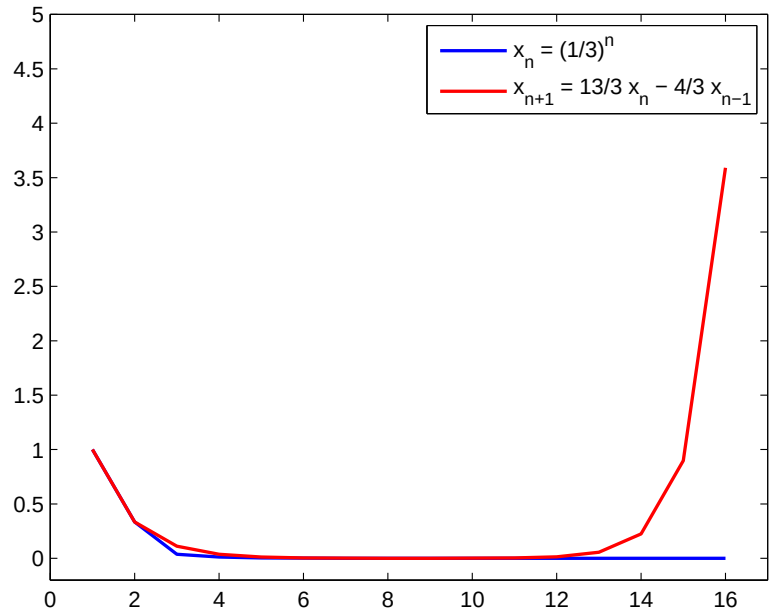
Algorytm numeryczny nazywamy **niestabilnym** jeśli małe błędy popełnione na jakimś etapie obliczeń rosną w następnych etapach i poważnie zniekształcają ostateczne wyniki.

Stabilność algorytmu

Algorytm numeryczny nazywamy **niestabilnym** jeśli małe błędy popełnione na jakimś etapie obliczeń rosną w następnych etapach i poważnie zniekształcają ostateczne wyniki.

Przykład

Wynik obliczenia wartości ciągu:
$x_n = (1/3)^n$
za pomocą wzoru rekurencyjnego:
$x_{n+1} = 13/3 x_n - 4/3 x_{n-1}$
$x_0 = 1, \; x_1 = 1/3,$
(Obliczenia pojedynczej precyzji)



Uwarunkowanie algorytmu

Algorytm numeryczny jest **źle uwarunkowany** jeśli małe zmiany danych początkowych wywołują duże zmiany wyników.

Dla pewnych typów zadań można określić wskaźnik uwarunkowania, który charakteryzuje wpływ zaburzeń danych na otrzymane wyniki. Jeśli jest on duży to zadanie jest źle uwarunkowane.

Uwarunkowanie jest cechą samego zadania i nie jest związane z metodą jego rozwiązania.

Wskaźnik uwarunkowania macierzy

Macierz A nazywamy źle uwarunkowaną jeśli jej wyznacznik $det(A)$ jest bliski zeru. Wskaźnik uwarunkowania macierzy:

$$cond(A) = \|A\|\|A^{-1}\|.$$

Macierz jest źle uwarunkowana jeśli wskaźnik uwarunkowania jest duży.
W przeciwnym przypadku macierz jest dobrze uwarunkowana.

Matlab: cond()

Aksjomaty normy

Każdemu wektorowi $\underline{\boldsymbol{x}}$ i każdej macierzy A można przyporządkować liczbę $\|\underline{\boldsymbol{x}}\|$ (normę x) oraz $\|A\|$ (normę A).

Normy macierzy maja następujące własności:

- $\|A\| > 0, \quad A \neq 0; \quad \|A\| = 0, \quad A = 0$
- $\alpha\|A\| = |\alpha|\|A\|, \quad \alpha \in \mathbb{R}$
- $\|A + B\| \leqslant \|A\| + \|B\|$
- $\|AB\| \leqslant \|A\|\|B\|$

Normy na wektorach

Najczęściej stosowane normy wektora $\underline{\boldsymbol{x}} = (x_1,\ x_2,\ \ldots,\ x_n)^T$ w przestrzeni $\mathbb{R}^n$:

- Norma euklidesowa:

$$\|\underline{\boldsymbol{x}}\| = \|\underline{\boldsymbol{x}}\|_2 := \sqrt{\sum_{i=1}^{n} x_i^2}.$$

  Wektor $\underline{\mathbf{x}}$ mający normę $\|\underline{\boldsymbol{x}}\|_2 = 1$ nazywamy wektorem jednostkowym.

- Norma maksimum:

$$\|\underline{\boldsymbol{x}}\| = \|\underline{\boldsymbol{x}}\|_\infty := \max_{1 \leqslant i \leqslant n} |x_i|.$$

- Norma - suma wartości bezwzględnych współrzędnych:

$$\|\underline{\boldsymbol{x}}\| = \|\underline{\boldsymbol{x}}\|_1 := \sum_{i=1}^{n} |x_i|.$$

Normy na macierzach

Najczęściej stosowane normy macierzy:

- Norma spektralna:

$$\|A\| = \|A\|_2 := \sqrt{\lambda_{max}(A^T A)}.$$

$\lambda_{max}(A^T A)$ - największa wartość własna macierzy $A^T A$.

- Norma wierszowa:

$$\|A\| = \|A\|_\infty := \max_{1 \leqslant i \leqslant n} \sum_{j=1}^{n} |a_{ij}|.$$

- Norma kolumnowa:

$$\|A\| = \|A\|_1 := \max_{1 \leqslant j \leqslant n} \sum_{i=1}^{n} |a_{ij}|.$$

Matlab: norm()

Uwarunkowanie macierzy – prosty algorytm identyfikacji

Załóżmy, że w wyniku przeprowadzonego prostego algorytmu identyfikacji dla systemu statycznego opisanego równaniem:

$$y = a_1 x_1 + a_2 x_2$$

otrzymano następujące macierze:

$$X_* = \begin{pmatrix} 1000 & 12 \\ 0.016 & 0.0001 \end{pmatrix}$$

$$cond(X_*) = 1.0871 \cdot 10^7$$

$$Y_* = \begin{pmatrix} 0.001 & 0.002 \end{pmatrix}$$

Macierz szukanych parametrów będąca rozwiązaniem równania identyfikacji $A = Y_* \cdot X_*^{-1}$ wynosi:

$$\boxed{A = \begin{pmatrix} 0.0003 & -21.6087 \end{pmatrix}}$$

Uwarunkowanie macierzy – prosty algorytm identyfikacji

Dla tego samego systemu przeprowadzono ponownie prosty algorytm identyfikacji i otrzymano następujące macierze:

$$X_* = \begin{pmatrix} 1000 & 12 \\ 0.016 & 0.0002 \end{pmatrix}$$

$$Y_* = \begin{pmatrix} 0.001 & 0.002 \end{pmatrix}$$

Macierz szukanych parametrów w tym przypadku wynosi:

$$\boxed{A = \begin{pmatrix} -0.0040 & 248.5000 \end{pmatrix}}$$

Uwarunkowanie macierzy – prosty algorytm identyfikacji

Identyfikacja na podstawie prostego algorytmu identyfikacji systemu statycznego opisanego równaniem:

$$y = a_1 x_1 + a_2 x_2$$

Zestawienie wyników dla dwóch eksperymentów idntyfikacyjnych:

$$X_* = \begin{pmatrix} 1000 & 12 \\ 0.016 & 0.0001 \end{pmatrix} \qquad X_* = \begin{pmatrix} 1000 & 12 \\ 0.016 & 0.0002 \end{pmatrix}$$

$$Y_* = \begin{pmatrix} 0.001 & 0.002 \end{pmatrix} \qquad Y_* = \begin{pmatrix} 0.001 & 0.002 \end{pmatrix}$$

$$\boxed{A = \begin{pmatrix} 0.0003 & -21.6087 \end{pmatrix}} \qquad \boxed{A = \begin{pmatrix} -0.0040 & 248.5000 \end{pmatrix}}$$

Błąd całkowity

Na **błąd całkowity** obliczeń numerycznych składają się następujące rodzaje błędów:

Błąd całkowity

Na **błąd całkowity** obliczeń numerycznych składają się następujące rodzaje błędów:

1. błąd wejścia,

Błąd całkowity

Na **błąd całkowity** obliczeń numerycznych składają się następujące rodzaje błędów:

1. błąd wejścia,
   - błąd wejścia dla podstawowych działań arytmetycznych,

Błąd całkowity

Na **błąd całkowity** obliczeń numerycznych składają się następujące rodzaje błędów:

1. błąd wejścia,
   - błąd wejścia dla podstawowych działań arytmetycznych,
2. błąd metody,

Błąd całkowity

Na **błąd całkowity** obliczeń numerycznych składają się następujące rodzaje błędów:

1. błąd wejścia,
   - błąd wejścia dla podstawowych działań arytmetycznych,
2. błąd metody,
   - błąd urwania procedury iteracyjnej,

Błąd całkowity

Na **błąd całkowity** obliczeń numerycznych składają się następujące rodzaje błędów:

1. błąd wejścia,
   - błąd wejścia dla podstawowych działań arytmetycznych,
2. błąd metody,
   - błąd urwania procedury iteracyjnej,
   - błąd dyskredytacji,

Błąd całkowity

Na **błąd całkowity** obliczeń numerycznych składają się następujące rodzaje błędów:

1. błąd wejścia,
   - błąd wejścia dla podstawowych działań arytmetycznych,
2. błąd metody,
   - błąd urwania procedury iteracyjnej,
   - błąd dyskredytacji,
3. błąd zaokrąglenia.

Błąd wejścia

Błąd wejścia zależy od:

Błąd wejścia

Błąd wejścia zależy od:

- niedokładności pomiarów,

Błąd wejścia

Błąd wejścia zależy od:

- niedokładności pomiarów,
- przeoczeń, błędnych odczytów, błędów drukarskich,

Błąd wejścia

Błąd wejścia zależy od:

- niedokładności pomiarów,
- przeoczeń, błędnych odczytów, błędów drukarskich,
- brakiem możliwości zapisu dowolnej liczby rzeczywistej w postaci liczby maszynowej

Błąd wejścia dla podstawowych działań arytmetycznych

Błąd bezwzględny dla podstawowych działań arytmetycznych określają zależności:

$$\Delta(x \pm y) = \Delta(x) \pm \Delta(y)$$

$$\Delta(xy) = \Delta(x)y + x\Delta(y) + \Delta(x)\Delta(y)$$

$$\Delta\left(\frac{x}{y}\right) = \frac{1}{y}\Delta(x) - \frac{x}{y^2}\Delta(y) + R$$

Błąd wejścia dla podstawowych działań arytmetycznych

Błąd względny dla podstawowych działań arytmetycznych określają zależności:

$$\delta(x \pm y) = \frac{x\delta(x) \pm y\delta(y)}{x \pm y}$$

$$\delta(xy) = \delta(x) + \delta(y) + \delta(x)\delta(y)$$

$$\delta\left(\frac{x}{y}\right) = \delta(x) - \delta(y) + R$$

Błąd wejścia dla podstawowych działań arytmetycznych

Mały błąd względny danych wejściowych podczas mnożenia i dzielenia powoduje mały błąd względny wyniku.

Błąd wejścia dla podstawowych działań arytmetycznych

Mały błąd względny danych wejściowych podczas mnożenia i dzielenia powoduje mały błąd względny wyniku.

Mały błąd względny danych wejściowych podczas dodawania i odejmowania może stać się duży, jeśli:

$$|x \pm y| \ll |x| + |y|$$

Występuje wówczas niebezpieczeństwo kasowania pozycji.

Odejmowanie bliskich wielkości

Podczas odejmowania bliskich sobie liczb może wystąpić niebezpieczeństwo kasowania pozycji (zmniejszenia liczby cyfr znaczących). Sytuacji takich należy unikać.

Odejmowanie bliskich wielkości

Podczas odejmowania bliskich sobie liczb może wystąpić niebezpieczeństwo kasowania pozycji (zmniejszenia liczby cyfr znaczących). Sytuacji takich należy unikać.

Przykład

Odejmowanie dwóch liczb, dysponujemy pięciopozycyjnym przedstawieniem mantysy:
$0.1004 \cdot 10^3 - 0.9988 \cdot 10^2 \rightarrow 0.1004 \cdot 10^3 - 0.09988 \cdot 10^3 = 0.00052 \cdot 10^3 = 0.5200 \cdot 10^0$.
Na końcu mantysy pojawiły się dwa zera co oznacza utratę dwóch bitów znaczących.

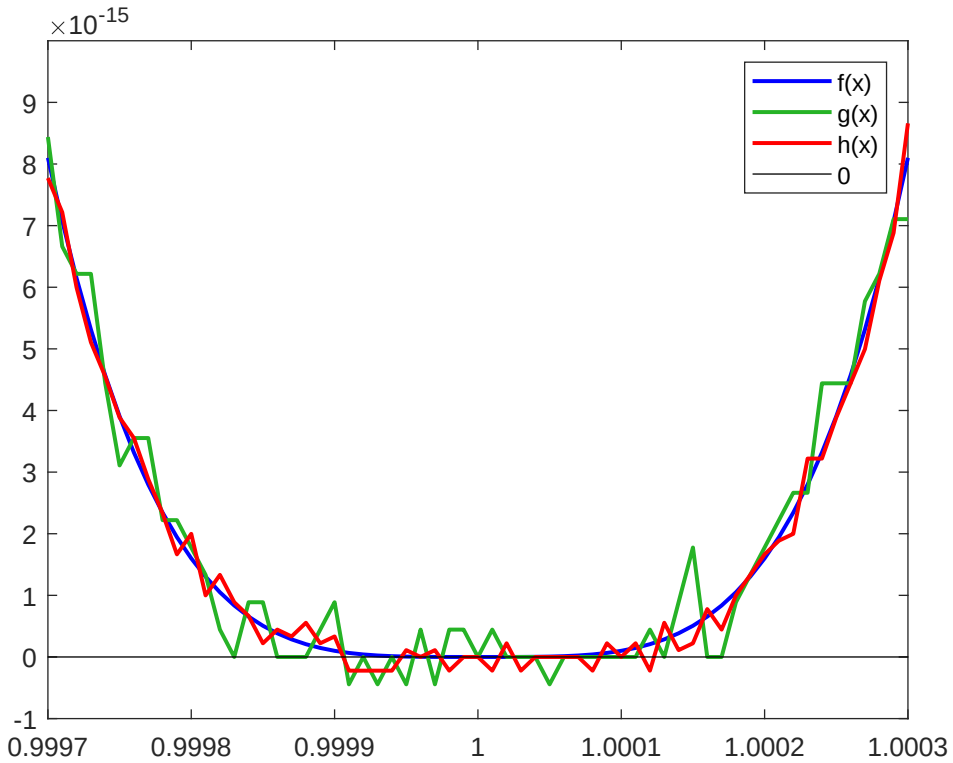
Dla małych x
$y = \sqrt{x^2+1} - 1 \quad \rightarrow \quad \frac{x^2}{\sqrt{x^2+1}+1}$

Odejmowanie bliskich wielkości

Przykład

Wynik obliczenia tej samej funkcji zapisanej pod różnymi postaciami w przedziale $< 0.9997, 1.0003 >$ z krokiem 0.00001.

$$f(x) = (x-1)^4$$
$$g(x) = x^4 - 4x^3 + 6x^2 - 4x + 1$$
$$h(x) = (((x-4)x+6)x-4)x+$$



Błąd metody

Błąd metody zależy od:

Błąd metody

Błąd metody zależy od:

- konieczności przybliżania wartości ciągłych,

Błąd metody

Błąd metody zależy od:

- konieczności przybliżania wartości ciągłych,
- nie zależy od błędów wejścia i zaokrąglenia

Błąd metody

Błąd metody zależy od:

- konieczności przybliżania wartości ciągłych,
- nie zależy od błędów wejścia i zaokrąglenia
- konieczności reprezentowania procesów nieskończonych

Błąd metody - błąd urwania procedury iteracyjnej

Na błąd urwania procedury iteracyjnej ma wpływ:

Błąd metody - błąd urwania procedury iteracyjnej

Na błąd urwania procedury iteracyjnej ma wpływ:

- kontrola liczby kroków iteracji, urwanie po określonej liczbie cykli iteracji, również wtedy gdy nie została jeszcze osiągnięta żądana dokładność,

Błąd metody - błąd urwania procedury iteracyjnej

Na błąd urwania procedury iteracyjnej ma wpływ:

- kontrola liczby kroków iteracji, urwanie po określonej liczbie cykli iteracji, również wtedy gdy nie została jeszcze osiągnięta żądana dokładność,
- śledzenie przebiegu rozwiązania na ekranie,

Błąd metody - błąd urwania procedury iteracyjnej

Na błąd urwania procedury iteracyjnej ma wpływ:

- kontrola liczby kroków iteracji, urwanie po określonej liczbie cykli iteracji, również wtedy gdy nie została jeszcze osiągnięta żądana dokładność,
- śledzenie przebiegu rozwiązania na ekranie,
- wykorzystanie znanych właściwości rozwiązania problemu,

Błąd metody - błąd urwania procedury iteracyjnej

Na błąd urwania procedury iteracyjnej ma wpływ:

- kontrola liczby kroków iteracji, urwanie po określonej liczbie cykli iteracji, również wtedy gdy nie została jeszcze osiągnięta żądana dokładność,
- śledzenie przebiegu rozwiązania na ekranie,
- wykorzystanie znanych właściwości rozwiązania problemu,
- zbadanie możliwości skalowania zmiennych lub funkcji,

Błąd metody - błąd urwania procedury iteracyjnej

Na błąd urwania procedury iteracyjnej ma wpływ:

- kontrola liczby kroków iteracji, urwanie po określonej liczbie cykli iteracji, również wtedy gdy nie została jeszcze osiągnięta żądana dokładność,
- śledzenie przebiegu rozwiązania na ekranie,
- wykorzystanie znanych właściwości rozwiązania problemu,
- zbadanie możliwości skalowania zmiennych lub funkcji,
- przeprowadzanie większej liczby testów, zmieniając długość kroku, warunek stopu, wartości startowe, itd.

Błąd metody - błąd dyskretyzacji

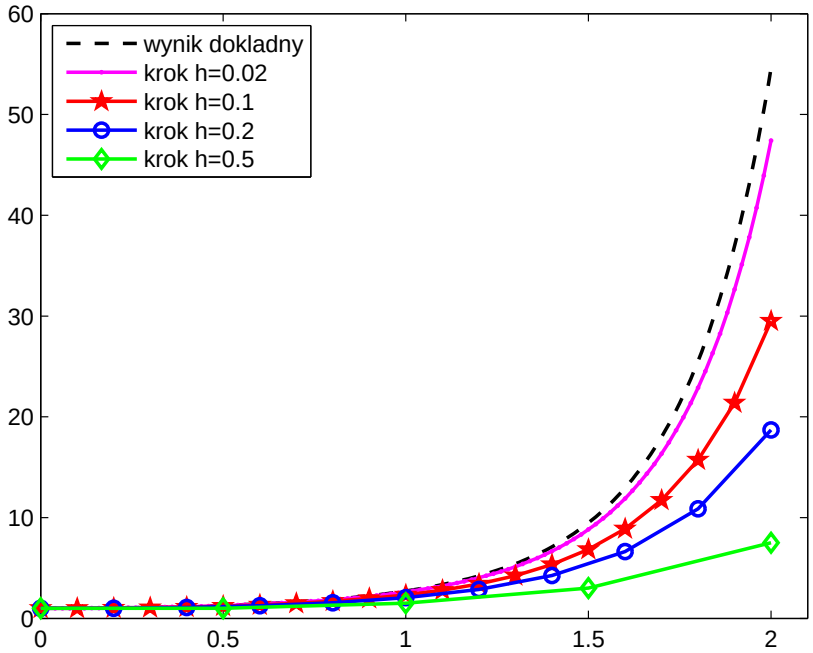
Błąd dyskretyzacji powstaje w wyniku aproksymacji struktur ciągłych za pomocą układów dyskretnych (np. całkowanie numeryczne).

Błąd metody - błąd dyskretyzacji

Błąd dyskretyzacji powstaje w wyniku aproksymacji struktur ciągłych za pomocą układów dyskretnych (np. całkowanie numeryczne).

Przykład

Wynik rozwiązania równania różniczkowego $\dot{y} = 2xy$ metodą Eulera dla różnych kroków dyskretyzacji.



Błąd zaokrąglenia

Nie każda liczba rzeczywista jest liczba maszynową, wobec tego trzeba zastąpić liczbę x jakąś bliska liczbą maszynową.

Błąd zaokrąglenia

Nie każda liczba rzeczywista jest liczba maszynową, wobec tego trzeba zastąpić liczbę x jakąś bliska liczbą maszynową.

Obcięcie

Jeśli bliska liczbie x liczba maszynowa $\mathrm{fl}(x)$ powstaje przez odrzucenie zbędnych bitów liczby x to mamy do czynienia z **obcięciem**.

Błąd zaokrąglenia

Nie każda liczba rzeczywista jest liczba maszynową, wobec tego trzeba zastąpić liczbę x jakąś bliska liczbą maszynową.

Obcięcie

Jeśli bliska liczbie x liczba maszynowa fl(x) powstaje przez odrzucenie zbędnych bitów liczby x to mamy do czynienia z **obcięciem**.

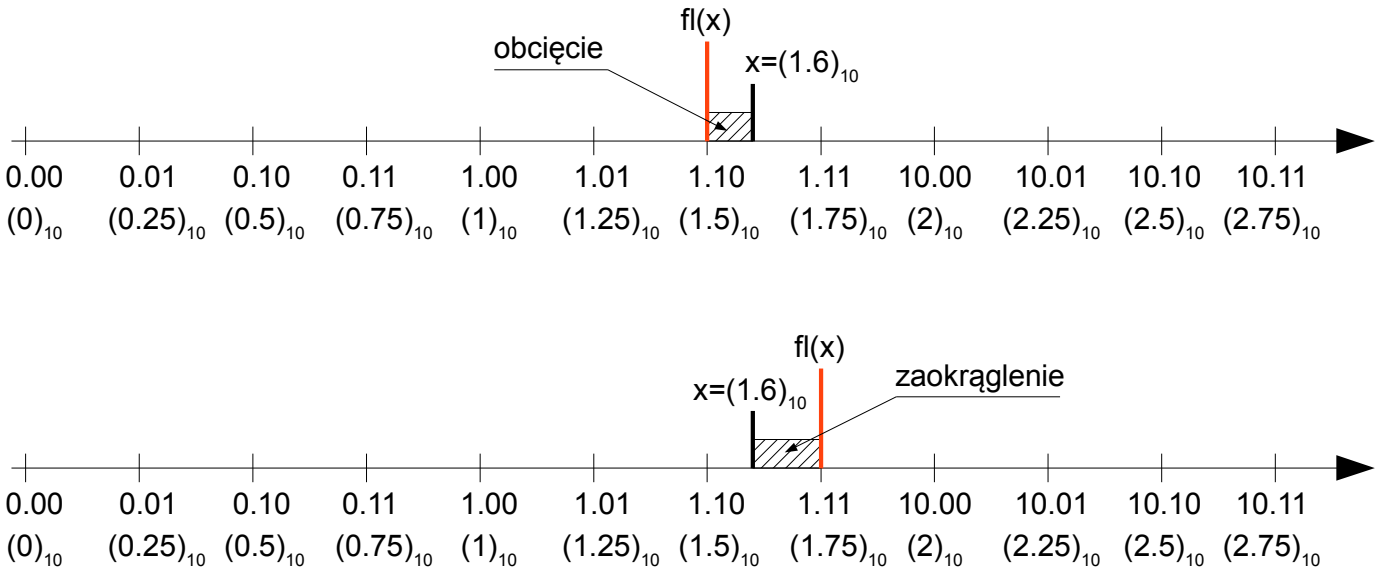
Zaokrąglenie

Jeśli inna bliska liczbie x liczba maszynowa fl(x) leżąca na prawo od x powstaje przez odrzucenie zbędnych bitów oraz dodaniu jedynki na ostatniej zachowanej pozycji to mamy do czynienia z **zaokrągleniem**.

Błąd zaokrąglenia

Przykład

Dysponujemy komputerem z 5 bitowym słowem (2 bity cecha, 2 bity mantysa, 1 bit znak) i chcemy zapisać w tym komputerze liczbę 1.6 ($(1.6)_{10} = (1.10011001...)_2$).
Wobec tego otrzymamy $fl(x) = (1.10)_2 = (1.5)_{10}$ (obcięcie) lub $fl(x) = (1.11)_2 = (1.75)_{10}$ (zaokrąglenie)



Błąd zaokrąglenia

Błąd zaokrąglenia powstaje również w trakcie operacji dodawania i odejmowania liczb o różnych rzędach.

Błąd zaokrąglenia

Błąd zaokrąglenia powstaje również w trakcie operacji dodawania i odejmowania liczb o różnych rzędach.

Aby dodać (odjąć) dwie liczby musimy wyrównać rzędy przesuwając mantysy o mniejszej wartości bezwzględnej.

Przykład

Dodawanie dwóch liczb, dysponujemy pięciopozycyjnym przedstawieniem mantysy.
$162.4 + 1.769 \rightarrow 0.1624 \cdot 10^3 + 0.1769 \cdot 10^1 \rightarrow 0.1624 \cdot 10^3 + 0.0017 \cdot 10^3 = 0.1641 \cdot 10^3$.
Wynik dokładny: $0.164169 \cdot 10^3$

Redukcja cyfr - sin

Problem: obliczanie wartości funkcji na podstawie rozwinięcia w szereg o wyrazach o różnych znakach na przykładzie funkcji sin.

$$sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

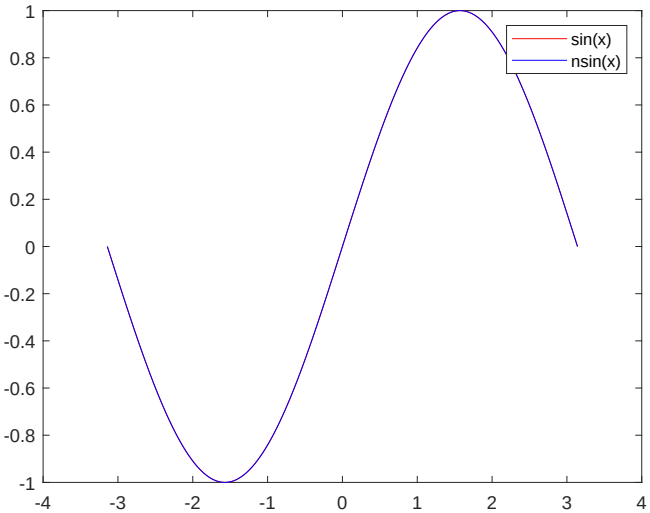
Algorytm w Matlabie

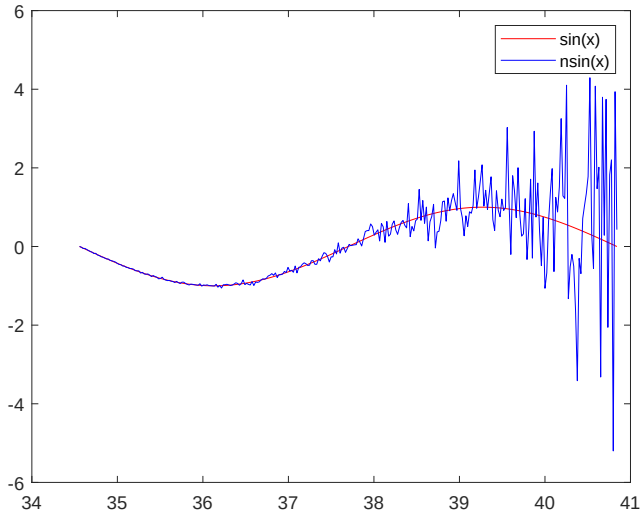
$function\ y = naszsin(x, N)$
$kx = -x. * x;$
$y = x;$
$for\ i = 1 : N$
    $x = x. * kx./((2 * k). * (2 * k + 1));$
    $y = y + x;$
end

Redukcja cyfr - sin

Problem: obliczanie wartości funkcji na podstawie rozwinięcia w szereg o wyrazach o różnych znakach na przykładzie funkcji sin.

- dla przedziału $(-pi, pi)$, i $N = 300$
- dla przedziału $(11 \cdot pi, 13 \cdot pi)$, i $N = 300$





Sumowanie z poprawianiem

Problem: obliczanie sumy dużej liczby elementów szeregu, np. od 1 do 10^6

Algorytm Kahana

$x = 1:1:10^6$
$n = length(x)$
$S = x(1)$
$C = 0;$
for $i = 2:n$
  $Y = x(i) - C$
  $T = S + Y$
  $C = (T - S) - Y$
  $S = T$
end

Sumowanie z poprawianiem

Problem: obliczanie sumy dużej liczby elementów szeregu, np. od 1 do 10^6

Algorytm Gilla-Mollera

$x = 1:1:10^6$
$n = length(x)$
$S = 0$
$U = 0$
$P = 0$
for $i = 1:n$
  $S = U + x(i)$
  $P = U - S + x(i) + P$
  $U = S$
end
$S = S + P$

Iloczyn skalarny

Problem: prawa przemienności z tradycyjnej arytmetyki nie zawsze działają w arytmetyce numerycznej

$$X = [exp(1), -pi, sqrt(2), -psi(1), log10(2)]$$
$$Y = [1486.2497, 878366.9879, -22.37492, 4773714.647, 0.000185049]$$

- Matlab, wynik $= 3.683013346744701e-04$
- od początku, wynik $= 3.683013935308729e-04$
- od końca, wynik $= 3.683011968860228e-04$
- kombinowany, wynik $= 3.683012910187244e-04$

Numeryczne rozwiązywanie zadań

Podczas numerycznego rozwiązywania danego zadania należy odpowiedzieć na następujące pytania:

Numeryczne rozwiązywanie zadań

Podczas numerycznego rozwiązywania danego zadania należy odpowiedzieć na następujące pytania:

- W obszarze danych.

Numeryczne rozwiązywanie zadań

Podczas numerycznego rozwiązywania danego zadania należy odpowiedzieć na następujące pytania:

- W obszarze danych.
  - Jakie wielkości są danymi?

Numeryczne rozwiązywanie zadań

Podczas numerycznego rozwiązywania danego zadania należy odpowiedzieć na następujące pytania:

- W obszarze danych.
  - Jakie wielkości są danymi?
  - Jakie przestrzenie danych i wyników najlepiej odpowiadają sensowi fizycznemu lub technicznemu?

Numeryczne rozwiązywanie zadań

Podczas numerycznego rozwiązywania danego zadania należy odpowiedzieć na następujące pytania:

- W obszarze danych.
  - Jakie wielkości są danymi?
  - Jakie przestrzenie danych i wyników najlepiej odpowiadają sensowi fizycznemu lub technicznemu?
- W obszarze uwarunkowania zadania.

Numeryczne rozwiązywanie zadań

Podczas numerycznego rozwiązywania danego zadania należy odpowiedzieć na następujące pytania:

- W obszarze danych.
  - Jakie wielkości są danymi?
  - Jakie przestrzenie danych i wyników najlepiej odpowiadają sensowi fizycznemu lub technicznemu?
- W obszarze uwarunkowania zadania.
  - Czy zadanie nie jest zbyt wrażliwe na zaburzenia danych?

Numeryczne rozwiązywanie zadań

Podczas numerycznego rozwiązywania danego zadania należy odpowiedzieć na następujące pytania:

- W obszarze danych.
  - Jakie wielkości są danymi?
  - Jakie przestrzenie danych i wyników najlepiej odpowiadają sensowi fizycznemu lub technicznemu?
- W obszarze uwarunkowania zadania.
  - Czy zadanie nie jest zbyt wrażliwe na zaburzenia danych?
  - Czy w danej arytmetyce można je rozwiązać?

Numeryczne rozwiązywanie zadań

Podczas numerycznego rozwiązywania danego zadania należy odpowiedzieć na następujące pytania:

- W obszarze danych.
  - Jakie wielkości są danymi?
  - Jakie przestrzenie danych i wyników najlepiej odpowiadają sensowi fizycznemu lub technicznemu?
- W obszarze uwarunkowania zadania.
  - Czy zadanie nie jest zbyt wrażliwe na zaburzenia danych?
  - Czy w danej arytmetyce można je rozwiązać?
  - Czy istnieje zadanie równoważne ale lepiej uwarunkowane?

Numeryczne rozwiązywanie zadań

Podczas numerycznego rozwiązywania danego zadania należy odpowiedzieć na następujące pytania:

Numeryczne rozwiązywanie zadań

Podczas numerycznego rozwiązywania danego zadania należy odpowiedzieć na następujące pytania:

- W obszarze jakości stosowanego algorytmu.

Numeryczne rozwiązywanie zadań

Podczas numerycznego rozwiązywania danego zadania należy odpowiedzieć na następujące pytania:

- W obszarze jakości stosowanego algorytmu.
  - Czy jest numerycznie stabilny i poprawny?

Numeryczne rozwiązywanie zadań

Podczas numerycznego rozwiązywania danego zadania należy odpowiedzieć na następujące pytania:

- W obszarze jakości stosowanego algorytmu.
  - Czy jest numerycznie stabilny i poprawny?
  - Jak duża może być utrata dokładności?

Numeryczne rozwiązywanie zadań

Podczas numerycznego rozwiązywania danego zadania należy odpowiedzieć na następujące pytania:

- W obszarze jakości stosowanego algorytmu.
  - Czy jest numerycznie stabilny i poprawny?
  - Jak duża może być utrata dokładności?
- W obszarze efektywności stosowanej metody.

Numeryczne rozwiązywanie zadań

Podczas numerycznego rozwiązywania danego zadania należy odpowiedzieć na następujące pytania:

- W obszarze jakości stosowanego algorytmu.
  - Czy jest numerycznie stabilny i poprawny?
  - Jak duża może być utrata dokładności?
- W obszarze efektywności stosowanej metody.
  - Co wiemy o złożoności obliczeniowej zadania?

Numeryczne rozwiązywanie zadań

Podczas numerycznego rozwiązywania danego zadania należy odpowiedzieć na następujące pytania:

- W obszarze jakości stosowanego algorytmu.
  - Czy jest numerycznie stabilny i poprawny?
  - Jak duża może być utrata dokładności?
- W obszarze efektywności stosowanej metody.
  - Co wiemy o złożoności obliczeniowej zadania?
  - Jaka jest efektywność innych metod?

Numeryczne rozwiązywanie zadań

Podczas numerycznego rozwiązywania danego zadania należy odpowiedzieć na następujące pytania:

- W obszarze jakości stosowanego algorytmu.
  - Czy jest numerycznie stabilny i poprawny?
  - Jak duża może być utrata dokładności?
- W obszarze efektywności stosowanej metody.
  - Co wiemy o złożoności obliczeniowej zadania?
  - Jaka jest efektywność innych metod?
  - Czy wybrana metoda jest optymalna (według przyjętego kryterium)?