Численные методы

Введение

Шишкин Владимир Андреевич

Пермский государственный национальный исследовательский университет

Структура курса

1. Введение
2. Приближение функций
   Интерполяция, аппроксимация, численное дифференцирование
3. Численное интегрирование
4. Задачи линейной алгебры
   Решение систем линейных алгебраических уравнений, проблема собственных значений
5. Нелинейные уравнения
6. Обыкновенные дифференциальные уравнения
   Задача Коши, краевые задачи
7. Численная оптимизация
   Безусловная и условная оптимизация, методы теории искусственного интеллекта

Литература

1. Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. *Численные методы* — М.: Лаборатория Базовых Знаний, 2000. — 624 с.
2. Бабенко К. И. *Основы численного анализа* — Москва-Ижевск: НИЦ «Регулярная и хаотическая динамика», 2002. — 848 с.
3. Мэтьюз Дж. Г., Финк К. Д. *Численные методы. Использование MATLAB* — 3-е изд. — М.: Изд. дом «Вильямс», 2001. — 720 с.
4. Алексеев Е. Р., Чеснокова О. В., Рудченко Е. А. *Scilab: Решение инженерных и математических задач* — М.: ALT Linux; БИНОМ. Лаборатория знаний, 2008. — 269 с.
5. Quarteroni A., Saleri F., Gervasio P. *Scientific Computing with MATLAB and Octave* — 3rd ed. — Springer, 2010. — 360 pp.

1. Исследование исходной задачи: P_1
2. Переход к упрощённой задаче: P_2
3. Построение программы: P_3
4. Решение задачи P_3
5. Анализ полученного решения

Метод вычислений

$\langle Q, I, \Omega, f \rangle$

Q — состояние вычислений;

I — ввод ($I \subset Q$);

Ω — вывод ($\Omega \subset Q$; $f(q) = q$ для всех $q \in \Omega$);

f — правило вычислений:

$$x_0 = x, \qquad x_{k+1} = f(x_k), \quad k \geqslant 0.$$

Вычисляемая последовательность $x_0, x_1, \ldots, x_k, \ldots$ *заканчивается через* k *шагов*, если k — это наименьшее целое число, для которого $x_k \in \Omega$.

Понятие алгоритма

Алгоритм

Метод вычислений, который заканчивается через конечное число шагов для всех x из I.

Свойства алгоритма:

1. конечность;
2. определённость;
3. ввод и вывод;
4. эффективность.

Погрешность результата численного решения

Причины погрешности:

1. неточное математическое описание задачи:
   - неточность числовых данных задачи (*неустранимая погрешность*);
   - несоответствие математического описания задачи реальности (*погрешность математической модели*);
2. неточный метод решения (*погрешность метода*);
3. ошибки округления при вводе данных, их арифметической обработке и выдаче (*вычислительная погрешность*).

Нормализованная вычислительная система с плавающей запятой

$$x = \pm 0, a_1 \dots a_t \times b^e = \pm f \times b^e \qquad (1)$$

$b \in \mathbb{Z}$ — основание, t разрядов a_i по основанию b составляют дробную часть (мантиссу) числа, e — порядок числа.

$$b^{-1} \leqslant f < 1$$

$$m \leqslant e \leqslant M$$

Числа вида (1) составляют конечное множество $\mathbb{Q}_f$.

Погрешности округления

$x \star y$ — арифметическая операция над $x, y \in \mathbb{R}$
$\tilde{x} \circledast \tilde{y}$ — арифметическая операция над $\tilde{x}, \tilde{y} \in \mathbb{Q}_f$

$$\tilde{x} \circledast \tilde{y} = (x \star y)(1 + \delta), \quad |\delta| \leqslant u$$

u — единица округления (машинная точность)

Погрешности округления

Абсолютная ошибка

$$\varepsilon_{\text{абс}}(\tilde{x}) = |x - \tilde{x}|$$

Относительная ошибка

$$\varepsilon_{\text{отн}}(\tilde{x}) = \frac{|x - \tilde{x}|}{|x|}$$

Погрешности округления

Значашие цифры — первая ненулевая цифра и все последующие цифры.

Приближение $\tilde{x}$ к x имеет p *правильных значащих цифр*, если $\tilde{x}$ и x округляются к одному числу с p значашими цифрами.

$$x = 0.9949, \quad \tilde{x} = 0.9951$$

Сколько правильных значащих цифр имеет $\tilde{x}$?

$$y = f(x), \quad \tilde{y} = f(x + \Delta x)$$

Прямой анализ (прямая трассировка)

$$x \to x + \Delta x \to f(x + \Delta x) \to \Delta y$$

Обратный анализ (обратная трассировка)

$$\Delta y \to \Delta x \to x + \Delta x$$

$$\tilde{y} + \Delta y = f(x + \Delta x), \qquad |\Delta y| \leqslant \epsilon |y|, \quad |\Delta x| \leqslant \eta |x|$$

Предполагая, что ϵ и η достаточно малы, говорят, что *вычисленное значение $\tilde{y}$ почти не отличается от значения $\tilde{y} + \Delta y$, которое получается при вводе $x + \Delta x$, которое почти не отличается от x.*

$\tilde{y}$ — почти правильный ответ для почти правильных данных.

Алгоритм называется вычислительно устойчивым, если малые изменения исходных данных влекут малое изменение решения.

Погрешности округления

$$y = f(x), \quad \tilde{y} = f(x + \Delta x), \qquad f \in \mathbf{C}_2$$

$$\tilde{y} - y = f(x + \Delta x) - f(x) = f'(x)\Delta x + \frac{f''(x + \theta\Delta x)}{2!}(\Delta x)^2, \quad \theta \in (0, 1)$$

$$\frac{\tilde{y} - y}{y} = \left(\frac{x f'(x)}{f(x)}\right)\frac{\Delta x}{x} + O\big((\Delta x)^2\big)$$

(Относительное) число обусловленности f

$$c(x) = \left|\frac{x f'(x)}{f(x)}\right|$$

прямая ошибка ≈ число обусловленности × обратная ошибка

Потеря значащих цифр

$$\lim_{x\to+0}\frac{1-\cos(x)}{x^2}=\frac{1}{2}$$

x	$\big(1-\cos(x)\big)/x^2$
10^{-1}	0.499583472197418
10^{-2}	0.499995833347366
10^{-3}	0.499999958325503
10^{-4}	0.499999996961265
10^{-5}	0.500000041370185
10^{-6}	0.500044450291170
10^{-7}	0.499600361081320
10^{-8}	0

Доказательные вычисления

- Интервальные вычисления: $x \in \mathbb{R}$ заменяется интервалом $[x_1, x_2]$, где $x_1, x_2 \in \mathbb{Q}_f$.

$$[x_1, x_2] + [y_1, y_2] = [x_1 + y_1, x_2 + y_2],$$
$$[x_1, x_2] - [y_1, y_2] = [x_1 - y_2, x_2 - y_1],$$
$$[x_1, x_2] \times [y_1, y_2] = [\min(x_1y_1, x_1y_2, x_2y_1, x_2y_2), \max(x_1y_1, x_1y_2, x_2y_1, x_2y_2)]$$
$$[x_1, x_2]/[y_1, y_2] = [x_1, x_2] \times \left[\frac{1}{y_2}, \frac{1}{y_1}\right], \text{ при } 0 \notin [y_1, y_2].$$

  При арифметических вычислениях применяется *направленное округление*.

- Использование рациональных чисел: a/b, где $a \in \mathbb{Z}$, $b \in \mathbb{N}$.

Программное обеспечение

- Языки программирования: Basic, C, C++, Pascal, C#, Fortran и т.д.
- Octave, Scilab, Matlab
- Maxima, Maple, Mathematica

$$x^3 = 2, \quad x = ?$$

Метод Ньютона:

$$x_0 = 1, \qquad x_{i+1} = \frac{2}{3}\left(x_i + \frac{1}{x_i^2}\right), \quad i = 0, 1, \ldots$$

Free Pascal:

```
program DQ;
var i : integer;
    x : double;
begin
  x := 1;
  for i := 1 to 10 do x := 2/3*(x+1/(x*x));
  writeln(x);
end.
```

C#:

```
using System;
namespace nummCS
{
  class Program
  {
    static void Main(string[] args)
    {
      double x = 1.0;
      for (int i = 0; i < 10; i++)
        x = 2.0 / 3.0 * (x + 1.0 / Math.Pow(x, 2));
      Console.WriteLine("x = {0}", x);
    }
  }
}
```

Пример

Задача удвоения куба

Octave:

```
x = 1;
for i=1:10
  x=2/3*(x+1/x^2);
endfor
x
```

Maxima:

```
x : 1$
for i: 1 thru 10 do
  x: 2/3*(x+1/x^2)$
done$
float(x);
```