MATLAB в научных исследованиях

Лекция № 7.

Обыкновенные дифференциальные уравнения

Н. Ю. Золотых

ННГУ, ВМК, сентябрь–декабрь 2008

Содержание

1. Что такое дифференциальное уравнение?

1. Что такое дифференциальное уравнение?

Пример. Предположим, что скорость роста некоторой популяции пропорциональна ее объему.

На коэффициент пропорциональности влияет сезон.

1. Что такое дифференциальное уравнение?

Пример. Предположим, что скорость роста некоторой популяции пропорциональна ее объему.

На коэффициент пропорциональности влияет сезон.

Пусть $u(t)$ — объем популяции.

1. Что такое дифференциальное уравнение?

Пример. Предположим, что скорость роста некоторой популяции пропорциональна ее объему.

На коэффициент пропорциональности влияет сезон.

Пусть $u(t)$ — объем популяции.

Приходим *к дифференциальному уравнению*

$$u' = (\alpha + \beta \sin \omega t)u.$$

1. Что такое дифференциальное уравнение?

Пример. Предположим, что скорость роста некоторой популяции пропорциональна ее объему.

На коэффициент пропорциональности влияет сезон.

Пусть $u(t)$ — объем популяции.

Приходим *к дифференциальному уравнению*

$$u' = (\alpha + \beta \sin \omega t)u.$$

Коэффициенты α, β, ω предполагаются известными.

1. Что такое дифференциальное уравнение?

Пример. Предположим, что скорость роста некоторой популяции пропорциональна ее объему.

На коэффициент пропорциональности влияет сезон.

Пусть $u(t)$ — объем популяции.

Приходим *к дифференциальному уравнению*

$$u' = (\alpha + \beta \sin \omega t)u.$$

Коэффициенты α, β, ω предполагаются известными.

Также известен объем популяции в начальный момент времени:

$$u(t_0) = u_0.$$

1. Что такое дифференциальное уравнение?

Пример. Предположим, что скорость роста некоторой популяции пропорциональна ее объему.

На коэффициент пропорциональности влияет сезон.

Пусть $u(t)$ — объем популяции.

Приходим *к дифференциальному уравнению*

$$u' = (\alpha + \beta \sin \omega t)u.$$

Коэффициенты α, β, ω предполагаются известными.

Также известен объем популяции в начальный момент времени:

$$u(t_0) = u_0.$$

Получили задачу: найти функцию $u(t)$, удовлетворяющую дифференциальному уравнению и начальному условию:

$$\begin{cases} u' = (\alpha + \beta \sin \omega t)u, \\ u(t_0) = u_0. \end{cases}$$

1. Что такое дифференциальное уравнение?

Пример. Предположим, что скорость роста некоторой популяции пропорциональна ее объему.

На коэффициент пропорциональности влияет сезон.

Пусть $u(t)$ — объем популяции.

Приходим *к дифференциальному уравнению*

$$u' = (\alpha + \beta \sin \omega t) u.$$

Коэффициенты α, β, ω предполагаются известными.

Также известен объем популяции в начальный момент времени:

$$u(t_0) = u_0.$$

Получили задачу: найти функцию $u(t)$, удовлетворяющую дифференциальному уравнению и начальному условию:

$$\begin{cases} u' = (\alpha + \beta \sin \omega t) u, \\ u(t_0) = u_0. \end{cases}$$

Это *задача Коши.*

1.1. Задача Коши

1.1. Задача Коши

Найти функцию $u = u(t)$, такую, что

$$u' = f(t, u), \quad u(t_0) = u_0,$$

1.1. Задача Коши

Найти функцию $u = u(t)$, такую, что

$$u' = f(t, u), \quad u(t_0) = u_0,$$

где f — заданная функция (от t и u),
t_0, u_0 — известные величины.

1.1. Задача Коши

Найти функцию $u = u(t)$, такую, что

$$u' = f(t, u), \quad u(t_0) = u_0,$$

где f — заданная функция (от t и u),
t_0, u_0 — известные величины.
$u' = f(t, u)$ — дифференциальное уравнение 1-го порядка,

1.1. Задача Коши

Найти функцию $u = u(t)$, такую, что

$$u' = f(t, u), \quad u(t_0) = u_0,$$

где f — заданная функция (от t и u),
t_0, u_0 — известные величины.

$u' = f(t, u)$ — дифференциальное уравнение 1-го порядка,

$u(t_0) = u_0$ — начальные условия.

1.1. Задача Коши

Найти функцию $u = u(t)$, такую, что

$$u' = f(t, u), \quad u(t_0) = u_0,$$

где f — заданная функция (от t и u),
t_0, u_0 — известные величины.

$u' = f(t, u)$ — дифференциальное уравнение 1-го порядка,

$u(t_0) = u_0$ — начальные условия.

Например,

1.1. Задача Коши

Найти функцию $u = u(t)$, такую, что

$$u' = f(t, u), \quad u(t_0) = u_0,$$

где f — заданная функция (от t и u),
t_0, u_0 — известные величины.

$u' = f(t, u)$ — дифференциальное уравнение 1-го порядка,

$u(t_0) = u_0$ — начальные условия.

Например,

$$u' = u, \quad u(0) = 1 \quad \Rightarrow \quad u = e^t$$

1.1. Задача Коши

Найти функцию $u = u(t)$, такую, что

$$u' = f(t, u), \quad u(t_0) = u_0,$$

где f — заданная функция (от t и u),
t_0, u_0 — известные величины.

$u' = f(t, u)$ — дифференциальное уравнение 1-го порядка,

$u(t_0) = u_0$ — начальные условия.

Например,

$$u' = u, \quad u(0) = 1 \quad \Rightarrow \quad u = e^t$$

$$u' = -10tu, \quad u(0) = 1 \quad \Rightarrow \quad u = e^{-5t^2}$$

1.1. Задача Коши

Найти функцию $u = u(t)$, такую, что

$$u' = f(t, u), \quad u(t_0) = u_0,$$

где f — заданная функция (от t и u),
t_0, u_0 — известные величины.

$u' = f(t, u)$ — дифференциальное уравнение 1-го порядка,

$u(t_0) = u_0$ — начальные условия.

Например,

$$u' = u, \quad u(0) = 1 \quad \Rightarrow \quad u = e^t$$

$$u' = -10tu, \quad u(0) = 1 \quad \Rightarrow \quad u = e^{-5t^2}$$

Поиск решения (аналитического или численного) называется *интегрированием* уравнения.

Поле направлений для уравнения $u' = -10tu$:

```
[T, U] = meshgrid(-1:.125:1);
F = -10*T.*U;
DT = 1./sqrt(1 + F.^2);
DU = F./sqrt(1 + F.^2);
quiver(T, U, DT, DU, 'b')
axis([-1, 1, -1, 1])
xlabel('t')
ylabel('u')
```

1.2. Системы дифференциальных уравнений

$$\begin{cases} u_1' = f_1(t, u_1, \ldots, u_n), \\ \dotfill \\ u_n' = f_n(t, u_1, \ldots, u_n) \end{cases} \quad \Leftrightarrow \quad u' = f(t, u), \quad \text{где} \quad u = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}.$$

1.2. Системы дифференциальных уравнений

$$\begin{cases} u_1' = f_1(t, u_1, \ldots, u_n), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ u_n' = f_n(t, u_1, \ldots, u_n) \end{cases} \quad \Leftrightarrow \quad u' = f(t, u), \quad \text{где} \quad u = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}.$$

Начальные условия (для задачи Коши):

$$\begin{cases} u_1(t_0) = u_{10}, \\ \ldots\ldots\ldots\ldots \\ u_n(t_0) = u_{n0} \end{cases} \quad \Leftrightarrow \quad u(t_0) = u_0.$$

1.3. Дифференциальные уравнения высших порядков

$$u^{(k)} = f(u^{(k-1)}, \ldots, u', u, t)$$

1.3. Дифференциальные уравнения высших порядков

$$u^{(k)} = f(u^{(k-1)}, \ldots, u', u, t)$$

Диф. ур. высшего порядка можно свести к диф. ур. 1-го порядка.

1.3. Дифференциальные уравнения высших порядков

$$u^{(k)} = f(u^{(k-1)}, \ldots, u', u, t)$$

Диф. ур. высшего порядка можно свести к диф. ур. 1-го порядка.

Например,

$$y'' + py' + qy = f(t)$$

1.3. Дифференциальные уравнения высших порядков

$$u^{(k)} = f(u^{(k-1)}, \ldots, u', u, t)$$

Диф. ур. высшего порядка можно свести к диф. ур. 1-го порядка.

Например,

$$y'' + py' + qy = f(t)$$

Пусть $u_1 = y$, $u_2 = y'$,

1.3. Дифференциальные уравнения высших порядков

$$u^{(k)} = f(u^{(k-1)}, \ldots, u', u, t)$$

Диф. ур. высшего порядка можно свести к диф. ур. 1-го порядка.

Например,

$$y'' + py' + qy = f(t)$$

Пусть $u_1 = y$, $u_2 = y'$,

тогда

$$\begin{cases} u_1' = u_2, \\ u_2' = y'' = f(t) - qu_1 - pu_2 \end{cases}$$

1.3. Дифференциальные уравнения высших порядков

$$u^{(k)} = f(u^{(k-1)}, \ldots, u', u, t)$$

Диф. ур. высшего порядка можно свести к диф. ур. 1-го порядка.

Например,

$$y'' + py' + qy = f(t)$$

Пусть $u_1 = y$, $u_2 = y'$,

тогда

$$\begin{cases} u_1' = u_2, \\ u_2' = y'' = f(t) - qu_1 - pu_2 \end{cases}$$

$$\begin{cases} u_1' = u_2, \\ u_2' = f(t) - qu_1 - pu_2 \end{cases}$$

2. MATLAB'овские решатели дифференциальных уравнений

//Далеко не все диф. уравнения можно решить аналитически.

```
ode45, ode23, ode113, ode15s, ode23s, ode23t, ode23tb

[t, U] = solver(odefun, tspan,  u0)
[t, U] = solver(odefun, tspan,  u0, options)
[t, U] = solver(odefun, tspan,  u0, options, P1, P2, ...)
   sol = solver(odefun, [t0, T], u0, ...)

    yy = deval(sol, t)
```

2.1. Примеры

2.1. Примеры

```
ode45(@fn, [0, 1], 1)
```

2.1. Примеры

```
ode45(@fn, [0, 1], 1)
```

```
ode45(@(t, u) -10*t*u, [0, 1], 1)
```

2.1. Примеры

```
ode45(@fn, [0, 1], 1)
```

```
ode45(@(t, u) -10*t*u, [0, 1], 1)
```

```
ode45(@(t, u, p) -p*t*u, [0, 1], 1, [], 10)
```

2.1. Примеры

```
ode45(@fn, [0, 1], 1)

ode45(@(t, u) -10*t*u, [0, 1], 1)

ode45(@(t, u, p) -p*t*u, [0, 1], 1, [], 10)

[t, u] = ode45(@(t, u) -10*t*u, [0, 1], 1)
```

2.1. Примеры

```
ode45(@fn, [0, 1], 1)
```

```
ode45(@(t, u) -10*t*u, [0, 1], 1)
```

```
ode45(@(t, u, p) -p*t*u, [0, 1], 1, [], 10)
```

```
[t, u] = ode45(@(t, u) -10*t*u, [0, 1], 1)
```

```
sol = ode45(@(t, u) -10*t*u, [0, 1], 1)
deval(sol, .33)
```

2.1. Примеры

```
ode45(@fn, [0, 1], 1)

ode45(@(t, u) -10*t*u, [0, 1], 1)

ode45(@(t, u, p) -p*t*u, [0, 1], 1, [], 10)

[t, u] = ode45(@(t, u) -10*t*u, [0, 1], 1)

sol = ode45(@(t, u) -10*t*u, [0, 1], 1)
deval(sol, .33)

tt = linspace(0, 1);
uu = deval(sol, tt);
plot(tt, uu)
```

2.2. Пример. Осциллятор

$$y'' + py' + qy = \sin(\omega t)$$

Положим

$$u_1 = y, u_2 = y'.$$

Получим систему

$$\begin{cases} u_1' = u_2, \\ u_2' = -qu_1 - pu_2 + \sin(\omega t) \end{cases}$$

См. osc.m

2.3. События

События связываются с обращением в нуль некоторых индикаторных выражений, например, одной из переменных. Также можно указать MATLAB'у, что нас интересуе некоторое событие, только если значение этого выражения убывает или, наоборот, возрастает.

2.3. События

События связываются с обращением в нуль некоторых индикаторных выражений, например, одной из переменных. Также можно указать MATLAB'у, что нас интересуе некоторое событие, только если значение этого выражения убывает или, наоборот, возрастает.

```
options = odeset('Events', eventsfunction);
[t, U, TE, UE, IE] = solver(odefun, tspan, u0, options);
```

2.3. События

События связываются с обращением в нуль некоторых индикаторных выражений, например, одной из переменных. Также можно указать MATLAB'у, что нас интересуе некоторое событие, только если значение этого выражения убывает или, наоборот, возрастает.

```
options = odeset('Events', eventsfunction);
[t, U, TE, UE, IE] = solver(odefun, tspan, u0, options);
```

где `TE`, `UE`, `IE` — время, значения переменных и номера наступивших событий

2.3. События

События связываются с обращением в нуль некоторых индикаторных выражений, например, одной из переменных. Также можно указать MATLAB'у, что нас интересуе некоторое событие, только если значение этого выражения убывает или, наоборот, возрастает.

```
options = odeset('Events', eventsfunction);
[t, U, TE, UE, IE] = solver(odefun, tspan, u0, options);
```

где `TE`, `UE`, `IE` — время, значения переменных и номера наступивших событий

`eventsfunction` — это функция вида

2.3. События

События связываются с обращением в нуль некоторых индикаторных выражений, например, одной из переменных. Также можно указать MATLAB'у, что нас интересуе некоторое событие, только если значение этого выражения убывает или, наоборот, возрастает.

```
options = odeset('Events', eventsfunction);
[t, U, TE, UE, IE] = solver(odefun, tspan, u0, options);
```

где `TE`, `UE`, `IE` — время, значения переменных и номера наступивших событий

`eventsfunction` — это функция вида

```
function [value, isterminal, direction] = eventsfunction(t, u)
```

2.3. События

События связываются с обращением в нуль некоторых индикаторных выражений, например, одной из переменных. Также можно указать MATLAB'у, что нас интересуе некоторое событие, только если значение этого выражения убывает или, наоборот, возрастает.

```
options = odeset('Events', eventsfunction);
[t, U, TE, UE, IE] = solver(odefun, tspan, u0, options);
```

где `TE`, `UE`, `IE` — время, значения переменных и номера наступивших событий

`eventsfunction` — это функция вида

```
function [value, isterminal, direction] = eventsfunction(t, u)
```

`value`, `isterminal`, `direction` — векторы длины m, где m — число событий.

2.3. События

События связываются с обращением в нуль некоторых индикаторных выражений, например, одной из переменных. Также можно указать MATLAB'у, что нас интересуе некоторое событие, только если значение этого выражения убывает или, наоборот, возрастает.

```
options = odeset('Events', eventsfunction);
[t, U, TE, UE, IE] = solver(odefun, tspan, u0, options);
```

где `TE`, `UE`, `IE` — время, значения переменных и номера наступивших событий

`eventsfunction` — это функция вида

```
function [value, isterminal, direction] = eventsfunction(t, u)
```

`value`, `isterminal`, `direction` — векторы длины m, где m — число событий.

в `value(j)` должно возвращаться значение j-го индикаторного выражения

2.3. События

События связываются с обращением в нуль некоторых индикаторных выражений, например, одной из переменных. Также можно указать MATLAB'у, что нас интересуе некоторое событие, только если значение этого выражения убывает или, наоборот, возрастает.

```
options = odeset('Events', eventsfunction);
[t, U, TE, UE, IE] = solver(odefun, tspan, u0, options);
```

где `TE`, `UE`, `IE` — время, значения переменных и номера наступивших событий

`eventsfunction` — это функция вида

```
function [value, isterminal, direction] = eventsfunction(t, u)
```

`value`, `isterminal`, `direction` — векторы длины m, где m — число событий.

в `value(j)` должно возвращаться значение j-го индикаторного выражения

в `isterminal(j)` должна возвращаться 1, если при наступлении j-го события процесс нахождения решения нужно прекратить, и 0 в противном случае

2.3. События

События связываются с обращением в нуль некоторых индикаторных выражений, например, одной из переменных. Также можно указать MATLAB'у, что нас интересуе некоторое событие, только если значение этого выражения убывает или, наоборот, возрастает.

```
options = odeset('Events', eventsfunction);
[t, U, TE, UE, IE] = solver(odefun, tspan, u0, options);
```

где `TE`, `UE`, `IE` — время, значения переменных и номера наступивших событий

`eventsfunction` — это функция вида

```
function [value, isterminal, direction] = eventsfunction(t, u)
```

`value`, `isterminal`, `direction` — векторы длины m, где m — число событий.

в `value(j)` должно возвращаться значение j-го индикаторного выражения

в `isterminal(j)` должна возвращаться 1, если при наступлении j-го события процесс нахождения решения нужно прекратить, и 0 в противном случае

в `direction(j)` должна возвращаться 1, -1 и 0, если событие наступает соответственно только при возрастании значения индикаторного выражения, его убывании или в любом случае

2.3.1. Пример. Прыжок с парашютом

2.3.1. Пример. Прыжок с парашютом

$$\begin{cases} my'' + ky|y| + mg = 0, \\ y(0) = y_0, \\ y'(0) = v_0. \end{cases}$$

2.3.1. Пример. Прыжок с парашютом

$$\begin{cases} my'' + ky|y| + mg = 0, \\ y(0) = y_0, \\ y'(0) = v_0. \end{cases}$$

Где k — коэффициент сопротивления воздуха.

2.3.1. Пример. Прыжок с парашютом

$$\begin{cases} my'' + ky|y| + mg = 0, \\ y(0) = y_0, \\ y'(0) = v_0. \end{cases}$$

Где k — коэффициент сопротивления воздуха.

С закрытым парашютом: $k = 0.24$. С открытым парашютом: $k = 45$

2.3.1. Пример. Прыжок с парашютом

$$\begin{cases} my'' + ky|y| + mg = 0, \\ y(0) = y_0, \\ y'(0) = v_0. \end{cases}$$

Где k — коэффициент сопротивления воздуха.

С закрытым парашютом: $k = 0.24$. С открытым парашютом: $k = 45$

Положим:

$$u_1 = y,$$
$$u_2 = y'.$$

Тогда

$$\begin{cases} u_1' = u_2, \\ u_2' = -ku_1|u_1|/m - g, \\ u_1(0) = y_0, \\ u_2(0) = v_0. \end{cases}$$

2.3.1. Пример. Прыжок с парашютом

$$\begin{cases} my'' + ky|y| + mg = 0, \\ y(0) = y_0, \\ y'(0) = v_0. \end{cases}$$

Где k — коэффициент сопротивления воздуха.

С закрытым парашютом: $k = 0.24$. С открытым парашютом: $k = 45$

Положим:

$$u_1 = y,$$
$$u_2 = y'.$$

Тогда

$$\begin{cases} u_1' = u_2, \\ u_2' = -ku_1|u_1|/m - g, \\ u_1(0) = y_0, \\ u_2(0) = v_0. \end{cases}$$

См. parachute.m

2.3.2. Пример. Полет тела, брошенного под углом к горизонту

2.3.2. Пример. Полет тела, брошенного под углом к горизонту

$$x'' = -kx'\sqrt{(x')^2 + (y')^2},$$
$$y'' = -ky'\sqrt{(x')^2 + (y')^2} - g,$$
$$x(0) = 0, \quad y(0) = h, \quad x'(0) = v\cos\alpha, \quad y'(0) = v\sin\alpha.$$

2.3.2. Пример. Полет тела, брошенного под углом к горизонту

$$x'' = -kx'\sqrt{(x')^2 + (y')^2},$$
$$y'' = -ky'\sqrt{(x')^2 + (y')^2} - g,$$
$$x(0) = 0, \quad y(0) = h, \quad x'(0) = v\cos\alpha, \quad y'(0) = v\sin\alpha.$$

Положим

$$u_1 = x, u_2 = x', u_3 = y, u_4 = y'$$

2.3.2. Пример. Полет тела, брошенного под углом к горизонту

$$\begin{aligned}
&x'' = -kx'\sqrt{(x')^2 + (y')^2},\\
&y'' = -ky'\sqrt{(x')^2 + (y')^2} - g,\\
&x(0) = 0, \quad y(0) = h, \quad x'(0) = v\cos\alpha, \quad y'(0) = v\sin\alpha.
\end{aligned}$$

Положим

$$u_1 = x, u_2 = x', u_3 = y, u_4 = y'$$

Тогда

$$\begin{aligned}
&u_1' = u_2,\\
&u_2' = -ku_2\sqrt{u_2^2 + u_4^2},\\
&u_3' = u_4,\\
&u_4' = -ku_4\sqrt{u_2^2 + u_4^2} - g,\\
&u_1(0) = 0, \quad u_3(0) = h,\\
&u_2(0) = v\cos\alpha, \quad u_4(0) = v\sin\alpha.
\end{aligned}$$

2.3.2. Пример. Полет тела, брошенного под углом к горизонту

$$
\begin{aligned}
&x'' = -kx'\sqrt{(x')^2 + (y')^2},\\
&y'' = -ky'\sqrt{(x')^2 + (y')^2} - g,\\
&x(0) = 0, \quad y(0) = h, \quad x'(0) = v\cos\alpha, \quad y'(0) = v\sin\alpha.
\end{aligned}
$$

Положим

$$u_1 = x, u_2 = x', u_3 = y, u_4 = y'$$

Тогда

$$
\begin{aligned}
&u_1' = u_2,\\
&u_2' = -ku_2\sqrt{u_2^2 + u_4^2},\\
&u_3' = u_4,\\
&u_4' = -ku_4\sqrt{u_2^2 + u_4^2} - g,\\
&u_1(0) = 0, \quad u_3(0) = h,\\
&u_2(0) = v\cos\alpha, \quad u_4(0) = v\sin\alpha.
\end{aligned}
$$

См. air.m

3. Extra slides

4. Устойчивость уравнения

$$\frac{\partial f}{\partial u} \begin{cases} < 0 & \text{уравнение устойчиво} \\ = 0 & \text{уравнение нейтрально устойчиво} \\ > 0 & \text{уравнение не устойчиво} \end{cases}$$

Уравнение $u' = -10tu$ обладает семейством решений $u = ce^{-5t^2}$

```
t = linspace(-1, 1)';
u = exp(-5*t.^2);
c = (0.04:0.01:0.1)*exp(5);
plot(t, u*c)
```

$\partial f/\partial u = -10t$

Для всех $t < 0$ задача не устойчива.

Для всех $t > 0$ задача устойчива.

4.1. Устойчивость системы

Якобиан

$$J(t, f_1, \ldots, f_n) = \begin{vmatrix} \frac{\partial f_1}{\partial u_1} & \ldots & \frac{\partial f_1}{\partial u_n} \\ \ldots & \ldots & \ldots \\ \frac{\partial f_n}{\partial u_1} & \ldots & \frac{\partial f_n}{\partial u_n} \end{vmatrix}$$

Если вещественные части всех собственных чисел якобиана меньше нуля, то система устойчива.

Если вещественная часть хотя бы одного собственного числа больше нуля, то система не устойчива.

Если вещественные части всех собственных чисел якобиана не больше нуля и вещественная часть по крайней мере одного собственного числа равна нулю, то система нейтрально устойчива.

5. Метод Эйлера

$$t_0 < t_1 < \ldots < t_n = T, \qquad u_j \approx u(t_j)$$

Например, равномерная сетка: $h = (T - t_0)/n$, $t_j = t_0 + j \cdot h$

$$u(t+h) = u(t) + h \underbrace{u'(t)}_{=f(t,u)} + h^2 \frac{u(\tau)}{2!} \tag{$\star$}$$

$$u_{j+1} = u_j + (t_{j+1} - t_j) f(t_j, u_j) \qquad u_{j+1} = u_j + h \cdot f(t_j, u_j)$$

odeeuler.m

5.1. Ошибка и порядок матода

Пусть $u_j(t)$ — точное решение задачи Коши с начальными условиями $u(x_j) = u_j$. В частности, $u(t) = u_0(t)$. Тогда

$$L_j = |u_j - u_{j-1}(t_j)| \quad (j = 1, 2, \ldots, n)$$

называется *локальной* ошибкой на j-м шаге, а

$$G = \max_j |u_j - u(t_j)|$$

— *глобальной* ошибкой. Если уравнение устойчиво, то глобальная ошибка меньше суммы локальных. Если уравнение не устойчиво, то глобальная ошибка может быть больше суммы локальных ошибок.

Если $L_j = O(h^{p+1})$, то для устойчивого уравнения

$$G < L_1 + L_2 + \ldots + L_n = \frac{T - t_0}{h} O(h^{p+1}) = O(h^p).$$

и p называется *порядком* метода.

Из формулы ($\star$) следует, что метод Эйлера имеет 1-й порядок.

6. Устойчивость метода

Для метода Эйлера

$$\left|1 + h \cdot \frac{\partial f}{\partial u}\right| < 1.$$

Поэтому если уравнение не устойчиво ($\partial f/\partial u > 0$), то и метод Эйлера не устойчив. Или, наоборот, уравнение *жесткое*, т. е. сильно устойчивое ($\partial f/\partial u \ll 0$), то метод Эйлера не устойчив.

Устойчивость метода Эйлера для систем дифференциальных уравнений:

$$\|1 + h \cdot J\| < 1.$$

7. Жесткие системы дифференциальных уравнений

Если

$$\frac{\partial f}{\partial u} \ll 0,$$

то уравнение называется *жестким*.

Если вещественные части некоторых собственных чисел якобиана J много меньше нуля, то система называется *жесткой*.

odeeulerdraw.m

8. Неявный метод Эйлера

$$u(t+h) = u(t) + h \underbrace{u'(t+h)}_{=f(t+h,u(t+h))} + h^2 \frac{u(\tau)}{2!} \qquad (\star$$

$$u_{j+1} = u_j + (t_{j+1} - t_j) f(t_{j+1}, u_{j+1}) \qquad u_{j+1} = u_j + h \cdot f(t_j, u_{j+1})$$

u_{j+1} может не выражать явно, тогда на каждом шаге необходимо решать уравнение.

odeeulerimplicit.m

odeeulerimplicitdraw.m

9. Системы линейных однородных дифференциальных уравнений

$$u' = Au, \qquad u(t_0) = u_0$$

Решение:

$$u = e^{A(t-t_0)} \cdot u_0 = \left(e^A\right)^{t-t_0} \cdot u_0$$

```
u = expm(A*(t-t0)) * u0
```

или

```
EA = expm(A);
u = EA^(t-t0) * u0
```

— для каждого t каждым из этих способов вычислять дорого.

Если $D = Q^{-1}AQ$, то

$$(t - t_0)A = Q((t - t_0)D)Q^{-1} \quad \text{и} \quad e^{(t-t_0)A} = Qe^{(t-t_0)D}Q^{-1}.$$

Если D — диагональная, то `expm(D) = exp(D)`.

```
[Q, D] = eig(A);
Q * exp((t - t0) * D) / Q
```

10. Краевая задача

$$u' = f(t, u), \quad u(a) = u_a, \quad u(b) = u_b$$

```
sol_init = bvpinit(t_init, u_init);
sol = bvp4c(@lhs, @border, sol_init);
uu = deval(sol, tt)
```

`sol.x`, `sol.y` — сеточная функция

Пример: bvp.m