

5.2. Съставяне на файл-функцията, пресмятаща производните на системата диференциални уравнения

5.2.1. Система диференциални уравнения от първи ред

Файл-функцията, пресмятаща производните на системата диференциални уравнения от първи ред (5.1), се оформя по следния начин:

```
function yp = fname(x,y)  
    % Name of the problem  
    yp = zeros(n,1) ;  
    yp(1) = f1(x,y(1),y(2),...,y(n)) ;  
    yp(2) = f2(x,y(1),y(2),...,y(n)) ;  
    .....  
    yp(n) = fn(x,y(1),y(2),...,y(n)) ;
```

Тук:

fname – име на файл-функцията;

x – независимата променлива (аргумента);

y – вектор, чиито елементи са функциите y_1, y_2, \dots, y_n ;

yp – вектор-стълб, чиито елементи са пресметнатите в тази функция първи производни y_1', y_2', \dots, y_n' .

Аргументите **x** и **y** са входящи, а аргументът **yp** – изходящ.

Забележки:

- Формалният аргумент **x** е задължителен дори и ако не присъства в изразите f_1, f_2, \dots, f_n ;
- При конкретна задача в десните части се кодират съответните изрази за първите производни;
- Числото **n**, представляващо броя на уравненията, е конкретна целочислена константа;
- Предварителното дефиниране размера на изходящия аргумент **yp** с помощта на оператора **yp = zeros(n,1)** повишава бързодействието. Не забравяйте, че той трябва да е вектор-стълб!

Възможен е и следният по-компактен вариант на файл-функцията:

```
function yp = fname(x,y)
    % Name of the problem
    yp = zeros(n,1);
    yp = [f1(x,y(1),y(2),...,y(n))
          f2(x,y(1),y(2),...,y(n))
          .....
          fn(x,y(1),y(2),...,y(n))] ;
```

5.2.2. Диференциално уравнение от n-ти ред

Файл-функцията за канонизираната система (5.4) можем да представим в следния компактен вид:

```
function yp = fname(x,y)
    % Name of the problem
    yp = zeros(n,1);
    yp(1:n-1) = y(2:n);
    yp(n) = f(x,y(1),y(2),...,y(n));
```

Възможен е и следният вариант на функцията:

```
function yp = fname(x,y)
    % Name of the problem
    yp = zeros(n,1);
    yp = [y(2);y(3);...;y(n)
          f(x,y(1),y(2),...,y(n))] ;
```



```

% Пресмятане на матрицата A
A = [a11 a12 ... a1N % aij = aij(t, y(1), y(2), ..., y(n))
      a21 a22 ... a2N % където y(1), y(2), ..., y(n)
      ..... % са неизвестните функции
      aN1 aN2 ... aNN];
% Пресмятане вектора-стълб на десните части b.
% В изразите на елементите на b, освен t, участвуват:
% y(1), y(2), ..., y(n) – функциите  $q_1, q_2, \dots, q_n$ 
% y(n+1), y(n+2), ..., y(2n) – първите производни на функциите
b = [b1(t, y(1), y(2), ..., y(2n))
      b2(t, y(1), y(2), ..., y(2n))
      .....
      bn(t, y(1), y(2), ..., y(2n))];
% Пресмятане на вторите производни (обобщените ускорения)
a = A\b; %  $a(1) = \ddot{q}_1, a(2) = \ddot{q}_2, \dots, a(n) = \ddot{q}_n$ 
% Пресмятане на първите производни  $\dot{y}_1, \dot{y}_2, \dots, \dot{y}_{2n}$ 
yt(1:n) = y(n+1:2n); %  $\dot{y}_1, \dot{y}_2, \dots, \dot{y}_n$ 
yt(n+1:2n) = a(1:n); %  $\dot{y}_{n+1}, \dot{y}_{n+2}, \dots, \dot{y}_{2n}$ 

```

При описване на някои процеси, например динамиката на механични системи със сухо триене във връзките, е възможно да се появят диференциални уравнения, нелинейни спрямо вторите производни (ускоренията):

$$(5.10) \quad f_j(\ddot{q}, \dot{q}, q, t) = 0, \quad j = 1, 2, \dots, n$$

Очевидно и в този случай, след прилагане на субституциите (5.6), можем да сведем системата (5.10) към диференциални уравнения от първи ред. Във файл-функцията, обаче, ще се наложи да решаваме *нелинейна* система алгебрични уравнения спрямо вторите производни. Това ще изисква съставянето на друга файл-функция, описваща уравненията (5.10). Видът на последната ще зависи от процедурата, която ще се използва за решаване на нелинейната система. Такъв тип задачи излизат извън обсега на нашето пособие.

5.2.5. Предаване на параметри от извикващата програма на файл-функцията

Обикновено, при изследване на дадени явления, се налага да решаваме многократно една задача за различни стойности на някои физически параметри. За да не коригираме всеки път файл-функцията е рационално, при написването ѝ, тези параметри да се кодират не като числа, а като променливи, които получават стойностите си от извикващата програма. Имената на тези параметри могат да бъдат

произволни, макар че в документацията на MATLAB те се обозначават с P1, P2, ..., PN. Предаването на стойностите на тези параметри от извикващата програма на файл-функцията може да стане по два начина:

- 1) Чрез обявяването им като глобални както в извикващата програма, така и във файл-функцията с помощта на оператора
global P1 P2 ...
- 2) Чрез включването им в списъка на аргументите на използваната за интегриране функция (Solver) и на файл-функцията:
function yt = fname(t, y, P1, P2, ...)

Препоръчваме Ви втория метод.

5.2.6. Програмно генериране на файл-функцията

Понякога се налага да решаваме не една частна задача, а цял клас от задачи, диференциалните уравнения на които се извеждат от някакъв общ принцип, теорема или уравнения. Пример за това могат да служат уравненията на Лагранж при динамично изследване на механични системи. В такъв случай е целесъобразно да използваме символните възможности на MATLAB (Symbolic Math Toolbox) за програмно извеждане на диференциалните уравнения и тяхното канонизиране.

Вместо да прекъсваме изпълнението на програмата за "ръчно" написване на файл-функцията и след това отново да продължим с численото интегриране, можем да се опитаме да генерираме програмно тази функция. По този начин ще обединим в една програма всички етапи от решението на задачата – извеждане на уравненията, канонизирането им, генерирането на файл-функцията, численото интегриране и визуализацията на резултатите (вижте следващата Глава 5).

Тук ще покажем как се създава програмно файл-функцията за случая 5.2.3 при $n = 2$. Ето как трябва да изглежда тя:

```
function yt = din2(t, y, p1)
  yt = zeros(4, 1);
  yt(1:2) = y(3:4);
  yt(3:4) = [f1(t, y(1), y(2), y(3), y(4), p1)
             f2(t, y(1), y(2), y(3), y(4), p1)];
```

Фрагментът от програмата, генерираща горната функция е:

```
% Генериране на файл-функцията din2.m
[Fid, mes] = fopen('din2.m', 'wt'); % отваряне на файла din2.m
fprintf(Fid, 'function yt = din2(t, y, p1) \n');
fprintf(Fid, ' yt = zeros(4,1); \n');
fprintf(Fid, ' yt(1:2) = y(3:4); \n');
fprintf(Fid, ' yt(3:4) = [f1(t, y(1), y(2), y(3), y(4), p1) \n');
fprintf(Fid, ' f2(t, y(1), y(2), y(3), y(4), p1)];');
fclose(Fid) % затваряне на файла din2.m
```

Пояснения:

- **Fid** е идентификатора на файла **din2.m**, създаван в момента на отварянето му с командата **open**. Той се използва от командата **fprintf** всеки път, когато искаме да напечатаме някакъв стринг в този файл;
- Опцията **'wt'** се използва при печат в нов файл или добавяне към съществуващ;
- Опцията **\n** заставя командата **fprintf** да печата следващата информация на нов ред;
- Файлът може да се използва едва след неговото затваряне с командата **close** ;
- След изпълнението на горния фрагмент, в текущата директория ще се появи файла **din2.m**, готов за използване при численото интегриране на системата. Моля проверете това, като отворите файла със системния редактор;
- За конкретна задача символичните функции **f1** и **f2** ще представяват конкретни изрази!