فصل ۹

کاربردها در محاسبات عددی

MATLAB An Introduction With Applications, 6th Edition
Dr. Amos Gilat
The Ohio State University

Slide deck by
Dr. Greg Reese
Miami University

روشهای عددی تکنیکهایی هستند برای محاسبه تقریبی پاسخ مسائل ریاضی. زمانی از روشهای عددی استفاده می شود که پاسخهای دقیق یا تحلیلی موجود نباشند یا به دست آوردن آنها غیر عملی باشد

این فصل به این مطالب می پردازد

- حل معادله با یک مجهول
- یافتن مقدار بیشینه یا کمینه یک تابع
- انتگرال گیری عددی
- حل معادلات دیفرانسیل معمولی مرتبه اول

یک معادله با یک متغیر (مستقل) را میتوان به این صورت نوشت: $f(x) = 0$

- جواب یا ریشه معادله مقداری برای متغیر است که در این معادله صدق کند، یعنی باعث شود $f(x)$ مقدار صفر داشته باشد
  - یافتن مقداری که در معادله صدق کند *ارضای* معادله نامیده میشود

از دید نموداری، جواب نقطه ای است که نمودار تابع $f(x)$ محور xها را قطع کند یا بر آن مماس شود

با این که از لحاظ ریاضی اگر تابع در مقدار x بر نمودار مماس شود ولی آن را قطع نکند آن مقدار ریشه محسوب میشود، در MATLAB یک عدد باید محل تقاطع نمودار با محور xها باشد تا به عنوان ریشه در نظر گرفته شود

معادله $f(x) = 0$ می تواند

- بدون جواب باشد
  - مثال $f(x) = 0$ هیچ جوابی ندارد اگر $f(x) = 5$
- تعداد محدودی جواب داشته باشد
  - مثال (یک جواب) $f(x) = 0$ دقیقاً یک جواب دارد اگر $f(x) = x$
  - مثال (دو جواب) $f(x) = 0$ دقیقاً دو جواب دارد اگر $f(x) = 1 - x^2$
- بی نهایت جواب داشته باشد
  - مثال $f(x) = 0$ بی نهایت جواب دارد اگر $f(x) = \sin(x)$

یک جواب یا ریشه عددی مقداری از x است که مقدار تابع را صفر یا تقریباً صفر کند

روند کلی یافتن جواب عددی عبارتست از

1. کاربر یک حدس اولیه برای ریشه عددی میزند، که روند از آن به عنوان تخمین حال حاضر ریشه عددی استفاده می کند
2. اگر تخمین حال حاضر مقدار تابع را تقریباً صفر کند، به عنوان ریشه عددی در نظر گرفته شده و محاسبه متوقف میشود
3. اگر تخمین حال حاضر مقدار تابع را تقریباً صفر نکند، حدس جدیدی زده شده که مقدار تابع را به صفر نزدیک تر کند و به گام دو می رود

- روند به کاربر اجازه تعریف «تقریباً صفر» را میدهد، مثلاً قدرمطلق کوچکتر از 10^{-6} باشد
- راه دیگر متوقف شدن روند این است که تخمینهای فعلی و قبلی تقریباً یکسان باشند
  - روند به کاربر اجازه تعریف «تقریباً یکسان» را میدهد، مثلاً قدرمطلق تفاضل کمتر از 10^{-4} باشد

از تابع `fzero()` برای یافتن ریشه یک تابع استفاده کنید

```
x = fzero(function,x0)
```

Solution

The function to be solved.

A value of *x* close to where the function crosses the axis.

- `fzero`یک تابع تابع است (قسمت ۹-۷ را ببینید) بنابراین یک تابع دیگر را به عنوان ورودی میگیرد (تابعی که باید حل شود)

جزئیات دیگری در خصوص متغیرهای fzero:

- x، حل عددی، یک عدد است
- functionتابعی است که باید حل شود. میتوان آن را به چند راه مختلف وارد کرد:
  - ساده ترین راه وارد کردن عبارت ریاضی به صورت رشته متنی است
  - یک تابع تعریف شده توسط کاربر در یک فایل بسازید و دستگیره آن را به عنوان ورودی بدهید (قسمت ۷.۹.۱)
  - یک تابع ناشناس بسازید (قسمت ۷.۸.۱) و سپس نام تابع ناشناس را به عنوان ورودی بدهید (قسمت ۷.۹.۱)

- باید function را به صورت استاندارد $f(x) = 0$ بنویسید. ممکن است لازم باشد تابع داده شده در مسأله را بازنویسی کنید، مثلاً
  - اگر $xe^{-x} = 0.2$ داده شده باشد، باید آن را به صورت $xe^{-x} - 0.2 = 0$ باز نویسی کرده و سپس تابع $f(x) = xe^{-x} - 0.2$ را به fzero بدهید
  - اگر بخواهید آن را به صورت یک رشته به fzero بدهید، به این صورت خواهد بود `'x*exp(-x)-0.2'`

- x0میتواند یک عدد یا بردار دو درایه ای باشد
  - اگر عدد باشد، باید مقداری از x در نزدیکی نقطه ای که تابع محور xها را قطع میکند باشد
  - اگر بردار باشد، دو درایه باید دو نقطه در دو طرف جواب باشند
    - این شرط ایجاب میکند که `f(x0(1))` و `f(x0(2))` علامتهای متفاوتی داشته باشند

- اگر یک تابع بیش از یک جواب داشته باشد، میتوانید جوابهای متفاوت را با استفاده از `fzero`، هر بار برای یک جواب با مقدار `x0` متفاوت بیابید
  - اگر تابع چند جمله ای باشد، استفاده از `roots` برای یافتن جوابها همه آنها را یک جا میدهد (قسمت ۸.۱.۲)
- یک راه مناسب برای به دست آوردن مقدار اولیه، رسم نمودار تابع و دیدن محل تقاطع نمودار با محور xها است

- در بسیاری از موارد میتوانید دامنه جواب را تخمین بزنید
- اغلب زمانی که یک تابع بیش از یک جواب داشته باشد تنها یکی از آنها معنای فیزیکی دارد

- fzeroصفرهای یک تابع را تنها زمانی که آنها محور xها را قطع کنند، و نه بر آن مماس شوند به دست میدهد
- اگر fzeroنتواند جوابی به دست آورد، NaN میدهد

```
>> fzero( @(x)x^2, 0.1 ) % touches at x=0
Exiting fzero: aborting search for an interval
containing a sign change because NaN or Inf
function value encountered during search.
(Function value at -1.37296e+154 is Inf.)
Check function or try again with a different
starting value.
ans =
   NaN
```

`fzero` گزینه های مختلفی دارد. برای دیدن همه آنها، `help fzero` را در پنجره فرمان وارد کنید یا از پنجره راهنمایی استفاده کنید

یک گزینه مفید

```
[x fval] = fzero(function,x0)
```

مقدار تابع `function` را در `x` می دهد. میتوانید این مقدار را به سادگی به دست آورید، ولی این حالت کمی در تایپ صرفه جویی می کند

مثال

سهمی با ریشه های ۰ و ۲

```
>> parabola = @(t)1 - (t-1)^2;
```

Find a zero

```
>> x = fzero( parabola, 4 )
x = 2
```

Get function value at the zero

```
>> parabola( x )
ans = 0
```

Do both at once

```
>> [x fval] = fzero( parabola, 4 )
x = 2
fval = 0
```

یک گزینه دیگر مقادیر استفاده شده حین محاسبه را نمایش میدهد

```
x = fzero( function, x0, optimset('display','iter') )
```

مثال

```
>> parabola = @(t)1 - (t-1)^2;
>> x = fzero( parabola, 4,...
optimset('display','iter') )
```

مثال – خروجی

```
Search for an interval around 4 containing a sign change:
 Func-count         a          f(a)             b          f(b)   Procedure
     1              4            -8             4            -8   initial interval
     3        3.88686      -7.33398       4.11314      -8.69162   search
     5           3.84       -7.0656          4.16       -8.9856   search
     7        3.77373      -6.69355       4.22627      -9.40885   search
     9           3.68       -6.1824          4.32      -10.0224   search
    11        3.54745      -5.48951       4.45255      -10.9201   search
    13           3.36       -4.5696          4.64      -12.2496   search
    15         3.0949      -3.38862        4.9051      -14.2498   search
    17           2.72       -1.9584          5.28      -17.3184   search
    19        2.18981      -0.41564       5.81019       -22.138   search
    20           1.44        0.8064       5.81019       -22.138   search
```

مثال – خروجی

```
Search for a zero in the interval [1.44, 5.81019]:
 Func-count           x          f(x)             Procedure
    20             1.44        0.8064             initial
    21          1.59359      0.647646             interpolation
    22          2.20238     -0.445718             interpolation
    23           1.9542     0.0894947             interpolation
    24           1.9957    0.00857678             interpolation
    25          2.00002  -3.57866e-005            interpolation
    26                2   7.70629e-008            interpolation
    27                2   6.89226e-013            interpolation
    28                2             0             interpolation

Zero found in the interval [1.44, 5.81019]
x =
     2
```

از `fzero` همچنین میتوان برای یافتن مقداری که به ازای آن تابع مقدار مشخصی داشته باشد استفاده کرد. فرض کنید $f(x)=0$ را داشته باشیم و مقداری از x را بخواهیم که به ازای آن $f(x)=3$. میتوان آن را به صورت $f(x)-3=0$ بازنویسی کرد و با تعریف $g(x) = f(x) - 3$ از `fzero` برای یافتن ریشه $g(x)$ استفاده کرد

مثال

مقداری برای x بیابید که به ازای آن معادله سهمی $1-(x-1)^2=0$ مقداری برابر ½ داشته باشد

$$(x = 1 \pm \frac{1}{\sqrt{2}})$$

```
>> parabola = @(x)1 - (x-1)^2;
>> g = @(x) parabola(x)-0.5;
>> fzero( g, 1 )
ans = 0.2929
>> 1 - 1/sqrt(2)
ans = 0.2929
```

اگر بخواهیم مقدار بیشینه یا کمینه تابع $y = f(x)$ را در ناحیه مشخصی از محور xها بیابیم، از این دستور استفاده میکنیم

```
[x fval] = fminbnd(function,x1,x2)
```

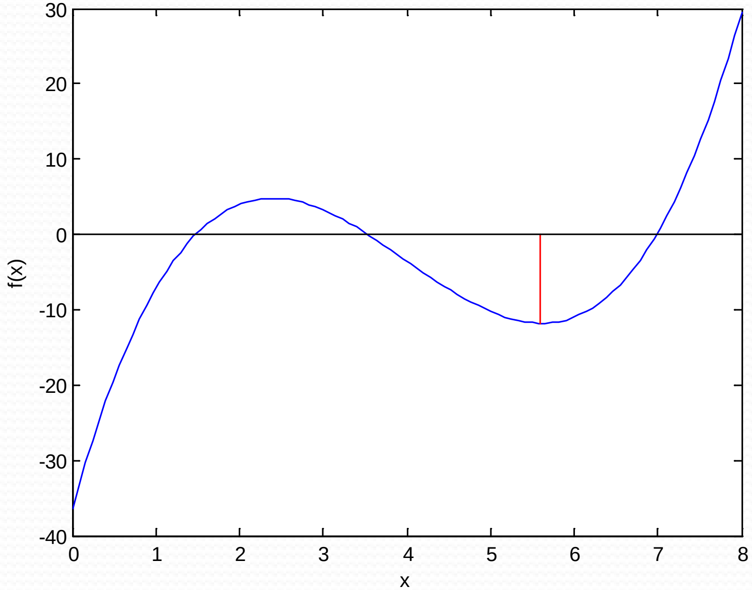
```
[x fval]= fminbnd(function,x1,x2)
```

- `function`–تابعی از یک متغیر مستقل x است
  - `function` می تواند یک رشته یا یک دستگیره باشد
  - `function` باید یک عدد x را بپذیرد
  - `function` باید یک عدد به عنوان خروجی بدهد – مقدار تابع در x
- `x1` و `x2` حد سمت چپ و راست بازه روی محور xها هستند به نحوی که `x1 < x2`

مثال

```
>> f=@(x)x^3 - 12*x^2 + 40.25*x - 36.5;
>> [x fval] = fminbnd( f, 3, 8 )
x = 5.6073
fval = -11.8043
>> [x fval] = fminbnd( f, 0, 8 )
x = 5.6073
fval = -11.8043
```

در کتاب جواب متفاوتی داده شده



برای یافتن مقدار بیشینه یک تابع، کمینه منفی تابع را به دست آورید

مثال – بیشینه $f(x) = xe^{-x} - 0.2$ را در بازه $0 \le x \le 8$ بیابید

```
>> f = @(x) x*exp(-x) - 0.2;
>> [x fval] = fminbnd( @(x)-f(x), 0, 8 )
x =
    1.0000
fval =
   -0.1679
```

بنابراین مقدار بیشینه 0.1679 است

حل تحلیلی بسیاری از انتگرالها غیر عملی یا غیر ممکن است. از MATLAB می توان برای تخمین مقدار تقریباً هر انتگرال معین $\int_a^b f(x)dx$ استفاده کرد. سه تابع برای انتگرال گیری عددی عبارتند از `quad`،`qual` و `trapz`

دستور quad:

```
q = quad(function,a,b)
```

The value of the integral. | The function to be integrated. | The integration limits.

- زمانی از آن استفاده کنید که بتوان تابع زیر انتگرال را به صورت یک تابع MATLAB نوشت
- از روش عددی «سیمپسون تطبیقی» استفاده می کند
- `(function)` *f(x)* یک بردار برای x می گیرد، بنابراین تابع داده شده باید مقدار هر درایه x را حساب کند
- باید مطمئن شد که تابع در بازه `[a,b]` مجانب قائم نداشته باشد

```
q = quad( function, a, b )
```

- تابع `function` را میتوان به صورت عبارت یا دستگیره تابع داد
  - مشابه `fzero` در قسمت ۹.۱
- `quad`انتگرال را با دقت مطلق کمتر از $1.0e{-6}$ حساب میکند
  - میتوان دقت را با اضافه کردن ورودی اختیاری `tol` به دستور به این صورت تغییر داد

    ```
    q = quad(function,a,b,tol)
    ```

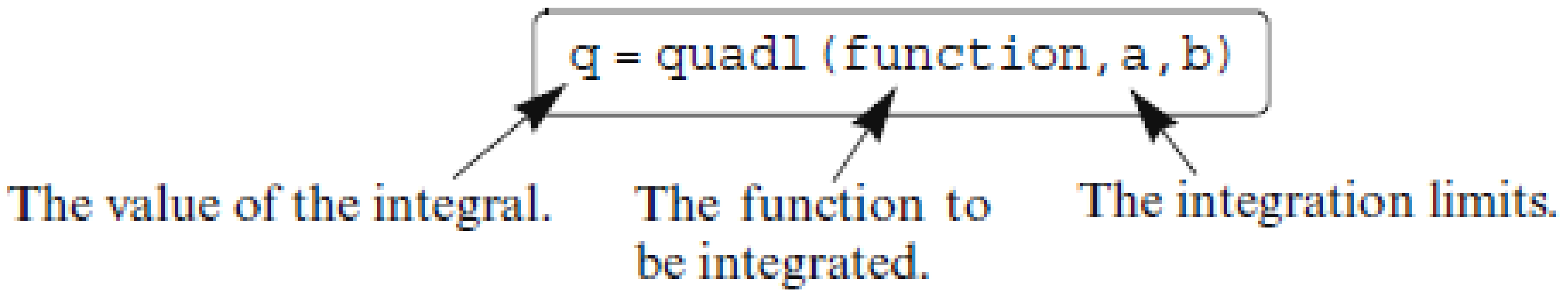
  - `tol`حداکثر خطا را تعیین میکند. مقادیر کوچکتر منجر به جوابهای دقیق تر میشود ولی نیاز به زمان محاسبه بیشتر دارد

مثال

از quad و یک تابع ناشناس برای محاسبه $\int_0^1 x^2 dx$ استفاده کنید (جواب ۱/۳ است)

```
>> parabola = @(x) x .* x
>> quad( parabola, 0, 1 )
ans =
    0.3333
```

دستور quadl:



- دقیقاً مشابه quad استفاده می شود
  - از روش عددی «لباتو تطبیقی»، که میتواند برای دقتهای بالاتر و انتگرالهای هموار مؤثرتر باشد استفاده میکند

مثال

از qual و یک عبارت رشته ای برای محاسبه $\int_0^1 x^2 dx$ استفاده کنید (جواب ۱/۳ است)

```
>> quad( 'x.*x', 0, 1 )
ans =
    0.3333
```

دستور `trapz:`

```
q = trapz( x, y )
```

- زمانی که نقاط (x,y) و نه تابع موجود باشند استفاده میشود
  - معمولاً در مورد داده های آزمایش به کار می رود
- از روش عددی انتگرال گیری ذوزنقه ای استفاده می کند
- x و y باید دارای طول برابر باشند
- مقادیر x باید به ترتیب صعودی باشند

مثال

از trapz و صد نقطه تصادفی برای محاسبه $\int_0^1 x^2 dx$ استفاده کنید (جواب ۱/۳ است)

```
>> x=sort( rand(1,100) );
>> y = x .* x;
>> trapz( x, y )
ans =
    0.3296
```

معادلات دیفرانسیل

- در علوم و مهندسی همه جا به کار می روند
- تنها تعداد محدودی از آنها را میتوان به صورت تحلیلی حل کرد
- تخمین عددی جواب تقریباً هر معادله دیفرانسیل را می توان به دست آورد

تنها معادلات دیفرانسیل معمولی مرتبه اول بررسی خواهند شد

- *مرتبه* – بالاترین مرتبه مشتق در معادله
  - *مرتبه اول* – تنها مشتق مرتبه اول وجود دارد
- *معمولی* – تنها یک متغیر مستقل وجود دارد
- *معادله دیفرانسیل* – معادله ای شامل یک تابع مجهول و مشتقات آن
- ODE – معادله دیفرانسیل معمولی

معادلات دیفرانسیل معمولی مرتبه اول
به این صورت هستند

$$\frac{dy}{dt} = f(t, y)$$

e.g.,

$$\frac{dy}{dt} = 5y \qquad \frac{dy}{dt} = ty^2$$

$$\frac{dy}{dt} = \sin(ty) + 20$$

در حالت کلی، بسیاری از توابع در یک معادله دیفرانسیل صدق میکنند

- به دنبال تابعی میگردیم که در معادله دیفرانسیل و مسأله خاص مورد بررسی صدق کند
- فرض میکنیم که مسأله مقدار متغیر وابسته y را در ابتدای دامنه متغیر مستقل t بدهد
  - جواب عددی را در بازه $t_0 \le t \le t_f$ یافته و از مقدار y در t_o یعنی $y_o = y(t_o)$ استفاده می کنیم

مسأله مقدار اولیه – یافتن تابعی که در یک معادله دیفرانسیل صدق کند و با مقدار داده شده در ابتدای بازه متغیر مستقل همخوانی داشته باشد

گامهای لازم برای حل یک معادله دیفرانسیل معمولی مرتبه اول:

گام ۱: مسأله را در صورت استاندارد بنویسید

معادله را به این صورت بنویسید

$\frac{dy}{dt} = f(t, y)$ برای $t_0 \le t \le t_f$, با $y = y_0$ در t_0

سه مورد اطلاعات مورد نیاز

1. معادله شامل مشتق مرتبه اول تابع
2. مقادیر متغیر مستقل
3. مقدار اولیه تابع

جواب مقدار y به عنوان تابعی از t است برای $t_o \le t \le t_f$

مثالی از یک مسأله مقدار اولیه

$$\frac{dy}{dt} = \frac{t^3 - 2y}{t}$$

برای $1 \le t \le 3$

با $y = 4.2$ در $t = 1$

گام ۲: یک تابع تعریف شده توسط کاربر (در یک فایل تابع) یا یک تابع ناشناس ایجاد کنید

تابعی برای سمت راست این معادله بنویسید

$$\frac{dy}{dt} = f(t, y)$$

مثلاً, `dydt = f( t, y )`

تابع باید ورودی هایی به ترتیب نوشته شده داشته باشد

- t – عددی به مقدار متغیر مستقل
- y – عددی به مقدار y در t، یعنی $y(t)$
- dydt – مشتق اول y در زمان t

مثال – برای

$$\frac{dy}{dt} = \frac{t^3 - 2y}{t}$$

فایل نمونه

```
function dydt = ODEexp1(t,y)
dydt = ( t^3 - 2*y ) / t;
```

تابع ناشناس نمونه

```
>> ode1 = @(t,y)(t^3 - 2*y)/t
```

گام ۳: یک روش حل انتخاب کنید

ایده کلی روشهای حل عددی مسائل مقدار اولیه

- فرض کنید $y(t)$ به اندازه کافی آرامٍ تغییر می کند که در فاصله کوچک Δt از t، $y(t)$ تقریبا یکسان با خط مماس در $y(t)$ باشد
- شیب m خط مماس گذرنده از $y(t)$ همان مشتق اول $y(t)$ است و میتوان آن را از این رابطه به دست آورد:

$$\frac{dy}{dt}(t)=f(t,y)$$

- معادله خط مماس $y=m{\cdot}t+b$ است و از آنجا که می دانیم این خط از $y(t)$ در t می گذرد، b را میتوان به دست آورد
- مقدار y را در زمان کوتاه Δt در آینده با این رابطه تخمین بزنید $y(t+\Delta t) = m{\cdot}(t+\Delta t)+b$
- این روند را در $t+2\Delta t$ تکرار کنید تا $y(t+3\Delta t)$ به دست آید
- روند را آنقدر ادامه دهید تا $t > t_f$

نهایتاً می رسید به

$$y(t_o),\ y(t_o+\Delta t)\ ,\ y(t_o+2\Delta t),\ldots,\ y(t_o+m\Delta t)$$

جدول ۱-۹ کتاب هفت عدد از حل کننده های ODE در MATLAB و برخی از ویژگی های آنها را معرفی کرده است

- *stiff problem* – یک ODE که پاسخ آن دارای قسمتهایی با تغییر کم در زمان و قسمتهایی با تغییر سریع است
- *one-step solver* – یک حل کننده که فقط از اطلاعات گام جاری برای به دست آوردن گام بعد استفاده می کند
- *multistep solver* – یک حل کننده که از اطلاعات گام جاری و گام های قبلی برای به دست آوردن گام بعدی استفاده می کند

توصیه کلی – ابتدا از ode45 استفاده کنید. اگر زمان زیادی طول کشید، ode15s را امتحان کنید

گام ۴: **معادله دیفرانسیل را حل کنید**

یکی از حل کننده های جدول ۹.۱ را این گونه فرا بخوانید

```
[t y] = solver_name(ODEfun,tspan,y0)
```

- `solver_name` – یکی از حل کننده های جدول ۹.۱ مثلاً `ode45, ode23t`
- `ODEfun` – یک تابع به صورتی که در گام ۲ توضیح داده شد
- `tspan` – یک بردار دو درایه ای از مقادیر اولیه و نهایی زمان، یعنی `[t0 tf]`
  - `tspan` میتواند بیش از دو درایه داشته باشد. راهنما را برای اطلاعات بیشتر ببینید

```
[t y]=solver_name(ODEfun,tspan,y0)
```

- y0 – مقدار اولیه y
- [t y]– دو بردار ستونی هم اندازه
  - tبردار نقاط زمانی است، به گونه ای که t(1)=t0 و t(end)=tf
  - yمقدار تابع در زمانهای متناظر در t است

$$\frac{dy}{dt} = \frac{t^3 - 2y}{t}$$ برای $1 \le t \le 3$ با $y = 4.2$ در $t = 1$

```
>> [t y]=ode45(@ODEexp1,[1:0.5:3],4.2)
t =
    1.0000
    1.5000
    2.0000
    2.5000
    3.0000
y =
    4.2000
    2.4528
    2.6000
    3.7650
    5.8444
```

The initial value.

The vector tspan.

The handle of the user-defined function ODEexp1.

شماره تمرین های منتخب

- ۶
- ۷
- ۹
- ۱۳
- ۱۵
- ۱۷
- ۱۹
- ۲۰
- ۲۲
- ۲۴

- ۲۵
- ۲۶
- ۲۷
- ۲۸
- ۳۲
- ۳۴
- ۳۶
- ۳۷
- ۴۰