一、Matlab 中多元方程组的表示方法

例 1、方程组 $\begin{cases}\sin x+y^2+\ln z-7=0\\ 3x+2^y-z^3+1=0\\ x+y+z-5=0\end{cases}$

上面方程组的左边是三个函数

$\begin{cases}F1(x,y,z)=\sin x+y^2+\ln z-7\\ F2(x,y,z)=3x+2^y-z^3+1\\ F3(x,y,z)=x+y+z-5\end{cases}$，方程组的解，就是这三个函数的零点。

如令 $X=(x,y,z)$，则 $X(1)$就是 x，$X(2)$就是 y，$X(3)$就是 z。

从而方程组也可以写为

$\begin{cases}F1=\sin(X(1))+X(2)^2+\ln X(3)-7\\ F2=3X(1)+2^{X(2)}-X(3)^3+1\\ F3=X(1)+X(2)+X(3)-5\end{cases}$

1、在 matlab 中，上面方程组的对应求解函数应如下设计：

```
function F=test(x)  % F 和 x 的记号可以随便。
  F(1)=sin(x(1))+x(2)^2+log(x(3))-7;
  F(2)=3*x(1)+2^x(2)-x(3)^3+1;
  F(3)=x(1)+x(2)+x(3)-5;
end
```

2、也可以采用下面方式以符合平时的使用习惯

```
function F=test(s)
x=s(1);y=s(2);z=s(3);
F(1)=sin(x)+y^2+log(z)-7; F(2)=3*x+2^y-z^3+1; F(3)=x+y+z-5;
End
```

3、也可以用匿名函数设计

```
>>f=@(x)[sin(x(1))+x(2)^2+log(x(3))-7;3*x(1)+2^x(2)-x(3).^3+1;x(1)+x(2)+x(3)-5]
```

二、求解单变量非线性方程方程 $f(x)=0$ 根的迭代法

（一）、牛顿迭代法

如果 $f(x)$连续可导，对初始点 x_0，函数的泰勒展开式为：

$$f(x)=f(x_0)+f'(x_0)(x-x_0)+\frac{f''(\xi)}{2}(x-x_0)^2$$

当某点 x_1 在 x_0 附近时 $|\Delta x|=|x_1-x_0|$ 很小，如果 $f''(x)$还有界，则

$$f(x_1)\approx f(x_0)+f'(x_0)(x_1-x_0),$$

如果 $f(x_0)+f'(x_0)(x_1-x_0)\approx 0$，那么 $f(x_1)\approx 0$，此时 x_1 就是函数的近似零点。

在 $f(x_0)+f'(x_0)(x_1-x_0)\approx 0$ 中解出 x_1，有 $x_1=x_0-\dfrac{f(x_0)}{f'(x_0)}$，

从而得递推公式 $x_{n+1}=x_n-\dfrac{f(x_n)}{f'(x_n)}$，当 $f(x_{n+1})=0$，或者 $f(x_{n+1})\approx 0$ 时，x_{n+1} 就是零点或者近似零点。

如果 $f(x_{n+1})\neq 0$ 或者不在 0 附近，那么就一直迭代下去。

（二）、不动点迭代法

1、原理：将方程 $f(x)=0$ 的形式写成 $x=\varphi(x)$，比如将 $x^2+2x=0$ 写成 $x=-\dfrac{x^2}{2}$。

对一个初始值 x_0，算出 $\varphi(x_0)$，令为 x_1，如果 $x_1=x_0$ 即 $\varphi(x_0)=x_0$ 或者 $\varphi(x_0)\approx x_0$，则说明 x_1 就是方程 $f(x)=0$ 的根或近似根；如果 x_1,x_0 差距比较大，就继续进行 $x_{n+1}=\varphi(x_n)$，直到 $x_{n+1}\approx x_n$ 为止。

2、使用此方法的条件

显然只有当数列 $x_0,x_1,x_2,\cdots,$ 收敛时，才能用这个方法找到根。所以设出的函数 $\varphi(x)$ 需满足一定的条件。

例如，求解 $f(x)=x^4-x-2$ 的根时，$\varphi(x)$ 的构造不唯一，可以构造函数 $x=\varphi(x)=x^4-2$ 或者 $x=\varphi(x)=\sqrt[4]{x^4+2}$ 或者 $x=\varphi(x)=\dfrac{2}{x^3-1}$。哪个构造产生的数列收敛呢？

定理 1：$\varphi(x)$ 在 $[a,b]$ 上具有连续的一阶导数，并且满足系列条件：

（1） 对任意的 $x\in[a,b]$，$\varphi(x)\in[a,b]$

（2） 对任意的 $x\in[a,b]$，$|\varphi'(x)|<1$

则有迭代法产生的数列收敛。

如果两个构造都收敛，那么哪个使用步骤最少呢？

定理 2：设迭代序列 $\{x_0,x_1,\cdots,x_n,\cdots\}$ 收敛于 x^*，记每次的迭代误差 $e(x_k)=x_k-x^*$，

若迭代误差 $\lim\limits_{k\to\infty}\dfrac{|e(x_{k+1})|}{|e(x_k)|^P}=c,(c\neq 0)$，即 $e(x_{k+1})$ 是 $e(x_k)$ 的 p 阶无穷小，此时称迭代过程 p 阶收敛，显然 p 越大，收敛越快，计算量越小。

三、求解多变量非线性线性方程组解的迭代法

n 个变量 n 个方程的方程组 $\begin{cases} f_1(x_1,x_2,\cdots,x_n)=0 \\ f_2(x_1,x_2,\cdots,x_n)=0 \\ \cdots \\ f_n(x_1,x_2,\cdots,x_n)=0 \end{cases}$

（一）、牛顿迭代法，

1、原理：

以二元函数构成的方程组为例 $\begin{cases} f(x,y)=0 \\ g(x,y)=0 \end{cases}$

假设函数 $f(x,y)$ 可微。对任给的初始点 (x_0,y_0)，如果 $f(x_0,y_0)\neq 0$，那么 (x_0,y_0) 不是 $f(x,y)=0$ 这个方程的根，此时在 (x_0,y_0) 附近找一个点 (x_1,y_1)，则由全微分知识可知

$$\Delta f = f(x_1,y_1)-f(x_0,y_0)=f_x'(x_0,y_0)\Delta x+f_y'(x_0,y_0)\Delta y+o(\rho)$$

其中 $\Delta x=x_1-x_0,\Delta y=y_1-y_0$，当 $\Delta x,\Delta y$ 很小时，$o(\rho)$ 忽略不计，于是可近似认为

$$f(x_1,y_1)-f(x_0,y_0)=f_x'(x_0,y_0)\Delta x+f_y'(x_0,y_0)\Delta y$$

如果 $f(x_1,y_1)=0$ 或者 $f(x_1,y_1)\approx 0$，那么 (x_1,y_1) 就是方程 $f(x,y)=0$ 的根或者近似根，代入上式有

$-f(x_0,y_0)=f_x'(x_0,y_0)\Delta x+f_y'(x_0,y_0)\Delta y$，写成矩阵的形式为：

$$-f(x_0,y_0)=(f_x',f_y')\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix}=(f_x',f_y')\begin{pmatrix}x_1-x_0\\ y_1-y_0\end{pmatrix}$$

类似的，有

$$-g(x_0,y_0)=(g_x',g_y')\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix}=(g_x',g_y')\begin{pmatrix}x_1-x_0\\ y_1-y_0\end{pmatrix}$$

将上面两式合写为矩阵的形式

$$-\begin{pmatrix}f(x_0,y_0)\\ g(x_0,y_0)\end{pmatrix}=\begin{pmatrix}f_x' & f_y'\\ g_x' & g_y'\end{pmatrix}\begin{pmatrix}x_1-x_0\\ y_1-y_0\end{pmatrix}=\begin{pmatrix}f_x' & f_y'\\ g_x' & g_y'\end{pmatrix}\begin{pmatrix}x_1\\ y_1\end{pmatrix}-\begin{pmatrix}f_x' & f_y'\\ g_x' & g_y'\end{pmatrix}\begin{pmatrix}x_0\\ y_0\end{pmatrix}$$

如记 $X_0=\begin{pmatrix}x_0\\ y_0\end{pmatrix},X_1=\begin{pmatrix}x_1\\ y_1\end{pmatrix},F(X_0)=\begin{pmatrix}f(x_0,y_0)\\ g(x_0,y_0)\end{pmatrix},F'(X_0)=\begin{pmatrix}f_x' & f_y'\\ g_x' & g_y'\end{pmatrix}$

则上式可简单记为 $-F(X_0)=F'(X_0)X_1-F'(X_0)X_0$，从而

$F'(X_0)X_1=F'(X_0)X_0-F(X_0)$，如果 $F'(X_0)$ 可逆，则在等式两边同时左乘以 $\left[F'(X_0)\right]^{-1}$ 可得

$X_1=X_0-\left[F'(X_0)\right]^{-1}F(X_0)$，即 $\begin{pmatrix}x_1\\ y_1\end{pmatrix}=\begin{pmatrix}x_0\\ y_0\end{pmatrix}-\begin{pmatrix}f_x' & f_y'\\ g_x' & g_y'\end{pmatrix}^{-1}\begin{pmatrix}f(x_0,y_0)\\ g(x_0,y_0)\end{pmatrix}$。

从上可以得到一个递推公式：$X_{n+1}=X_n-\left[F'\left(X_n\right)\right]^{-1}F\left(X_n\right)$

如果$F\left(X_{n+1}\right)\approx 0$则停止迭代，如果$F\left(X_{n+1}\right)\neq 0$，那么就一直迭代下去。

可以将上述讨论推广到方程组$\begin{cases}f_1\left(x_1,x_2,\cdots,x_n\right)=0\\ \cdots\\ f_2\left(x_1,x_2,\cdots,x_n\right)=0\end{cases}$。

（二）、不动点迭代法

1、原理：将方程组$\begin{cases}f_1\left(x_1,x_2,\cdots,x_n\right)=0\\ \cdots\\ f_2\left(x_1,x_2,\cdots,x_n\right)=0\end{cases}$写成$\begin{cases}x_1=\varphi_1\left(x_1,x_2,\cdots x_n\right)\\ x_2=\varphi_2\left(x_1,x_2,\cdots x_n\right)\\ \cdots\\ x_n=\varphi_n\left(x_1,x_2,\cdots x_n\right)\end{cases}$

与单变量不动点迭代法类似，$\varphi\left(x\right)$有多种设法

从而得到迭代公式$X_{n+1}=\varphi\left(X_n\right)$，如果$X_{n+1}=X_n$，说明$X_{n+1}$是方程得根，否则就一直迭代下去，直到$X_{n+1}\approx X_n$满足精度要求。

2、成立条件：设求解范围为D，$x,y\in D$

（1）$\left|\varphi\left(x\right)-\varphi\left(y\right)\right|\le L\left|x-y\right|$，$0<L<1$

（2）$\varphi\left(x\right)\in D$

四、常微分方程初值问题的数值解法

微分方程的数值解法的目的是想办法找出满足微分方程的一系列的数据点$\left(x_n,y_n\right)$，其思维是：设$\left(x_0,y_0\right)$是初始已知点，从初始点出发，给自变量x_0设置一个步长h，比如令$h=0.05$，则，下一个点的自变量值为$x_1=x_0+h$，如果能够想办法求出下一个点的因变量值y_1，就可以得到新的初始点$\left(x_1,y_1\right)$，依此类推，就可以得到曲线上的一系列点$\left(x_0,y_0\right),\left(x_1,y_1\right),\cdots$，然后用这些点拟合曲线，拟合曲线就可以作为方程的近似解。

下面介绍一阶微分方程的数值解法。

（一）已知$\dfrac{dy}{dx}=f\left(x,y\right)$，$y_0=y\left(x_0\right)$，求$y=y\left(x\right)$。

欧拉法求数据点：

设$\left(x_n,y_n\right)$是初始点，给x_n一个步长h，下一个点是$\left(x_{n+1},y_{n+1}\right)$，其中$y_{n+1}$待定，由$y'=f\left(x,y\right)$。

两边积分有，$\int_{x_n}^{x_{n+1}}y'dx=\int_{x_n}^{x_{n+1}}f\left(x,y\right)dx$，得$y\left(x_{n+1}\right)-y\left(x_n\right)=\int_{x_n}^{x_{n+1}}f\left(x,y\right)dx$为方便，记

$y_{n+1}=y(x_{n+1})$, $y_n=y(x_n)$，则 $y_{n+1}-y_n=\int_{x_n}^{x_{n+1}} f(x,y)dx$，右边积分是曲线 $f(x,y(x))$ 在区间段上的曲边梯形面积。由于 $\int_{x_n}^{x_{n+1}} f(x,y)dx=f(\xi,y(\xi))\cdot h$ （$x_n<\xi<x_{n+1}$）

如用下端点 $f(x_n,y_n)$ 替代 $f(\xi,y(\xi))$，则 $y_{n+1}\approx y_n+h\cdot f(x_n,y_n)$ （1）（欧拉法）

如用上端点 $f(x_{n+1},y_{n+1})$ 替代 $f(\xi,y(\xi))$，则 $y_{n+1}\approx y_n+h\cdot f(x_{n+1},y_{n+1})$ （2）（欧拉后退法）

公式（1）用前点值推后点值，只需将初始点 (x_0,y_0) 代入，即可逐步推出后点值的近似值。

注意：方程（1）（2）的左边 y_{n+1} 是函数真值，右边算出的值只是它的近似值。

公式（2）也可以推出后点值，但是个关于 y_{n+1} 的隐函数方程，想解出 y_{n+1} 不如方程（1）方便

（注，方程（2）求解 y_{n+1} 可以用迭代法求解，解法原理为，先用（1）求出一个 y_{n+1} 的估计值 y_{n+1}^0，将之代入（2），算出另一个 y_{n+1} 的估计值 y_{n+1}^1，如果 $|y_{n+1}^1-y_{n+1}^0|<tol$（容差），那么将 y_{n+1}^1 作为 y_{n+1}，否则就一直往下迭代，直到前后 y_{n+1} 的估计值的差距落在指定的容差内，此时将最后的 y_{n+1} 的估计值作为 y_{n+1}）。

也可以考虑用梯形面积替代曲边梯形面积，此时 $y_{n+1}\approx y_n+\frac{h}{2}\left[f(x_n,y_n)+f(x_{n+1},y_{n+1})\right]$ （3）

这也是一个关于 y_{n+1} 的隐函数方程，解出 y_{n+1} 的方法与方程（2）类似，也是用迭代法求解。

通常为了解出方便，会采用**改进的欧拉公式**进行求解，即先用公式（1）算出 y_{n+1} 的预测值 y_{n+1}^0，这个 y_{n+1}^0 不作为最终的 y_{n+1}，而是把它代入（3）的右边，再将算出的值作为 y_{n+1} 的最终估计值。即 $y_{n+1}^0=y_n+h\cdot f(x_n,y_n)$，代入（3）得到 $y_{n+1}\approx y_n+\frac{h}{2}\left[f(x_n,y_n)+f(x_{n+1},y_{n+1}^0)\right]$。每次求后面点的时候都这么做。

上面计算出来的 y_{n+1} 只是真实值 $y(x_{n+1})$ 的近似值，由于在推导后面的 y_{n+2} 时也会用到 y_{n+1} 的值，这会导致计算 y_{n+1} 产生的误差被代入到计算 y_{n+2} 中，这种情况从计算 y_1 时就开始了，所以计算某项 y_k 时的误差其实是包含了前面所有点的整体误差，为了减少误差和计算的便利性数学家们研究了多种方法，用的比较广泛的是龙格库塔方法。