数值分析第二次上机作业

宋歌 2015080086 数 52

10/30/2018

用不同的方法求解方程 $x^3+2x^2+10x-20=0$ 在 $x_0=1$ 附近的根。

1 实验原理

1.1 不动点迭代

不动点迭代法的基本思想是：将求解非线性方程的根转化为求解函数的不动点，即将求解 $f(x)=0$ 转化为求解 $\varphi(x)=x$。我们将 φ 作为迭代函数，希望在 $x_{k+1}=\varphi(x_k)$ 的迭代格式下，序列 $\{x_k\}$ 最终可以收敛到不动点 α。而压缩映射原理则保证了这样的迭代格式在某些限制条件下可以收敛。

压缩映射原理：设 $\varphi: D\subset\mathbb{R}^n\to\mathbb{R}^n$ 在闭集 $D_0\subset D$ 中是压缩映射，即存在 $C\in(0,1)$，使得 $\forall x,y\in D_0\subset D$，有 $||\varphi(x)-\varphi(y)||\leqslant C||x-y||$，且 $\varphi(D_0)\subset D_0$，则 $\varphi(x)$ 在 D_0 上存在唯一的不动点。

因此，若 α 为 $\varphi(x)=x$ 的不动点，$\varphi'(x)$ 在 α 邻域内连续且 $|\varphi'(\alpha)|<1$，则存在 $\delta>0$ 以及 $C\in(0,1)$ 使得当 $x\in[\alpha-\delta,\alpha+\delta]$ 时，$|\varphi'(x)|\leqslant C<1$，此时对于 $\forall x,y\in[\alpha-\delta,\alpha+\delta]$，我们有 $|\varphi(x)-\varphi(y)|=|\varphi'(\xi)|*|x-y|\leqslant C|x-y|$，即 φ 为压缩映射，从而迭代 $x_{k+1}=\varphi(x_k)$ 在 α 的邻域内具有局部收敛性。

1.2 Steffensen 加速

假设由迭代函数 $\varphi(x)$ 产生的序列 $\{x_k\}_{k=0}^{+\infty}\to\alpha=\varphi(\alpha)$，则由微分中值定理，存在 $\xi\in(x_0,\alpha)$ 使得：

$$x_1-\alpha=\varphi(x_0)-\varphi(\alpha)=\varphi'(\xi)(x_0-\alpha)$$

假设 $\varphi(x)$ 在 α 附近变化不大，即 $\varphi'(x)$ 在 α 附近约为一个常数 L，则有 $x_1-\alpha\approx L(x_0-\alpha), x_2-\alpha\approx L(x_1-\alpha)$，进一步得到 $\frac{x_2-\alpha}{x_1-\alpha}\approx\frac{x_1-\alpha}{x_0-\alpha}$，从而可以利用 x_1,x_2 进一步修正近似 x_0：

$$\alpha\approx\frac{x_0x_2-x_1^2}{x_2-2x_1+x_0}=x_0-\frac{(x_1-x_0)^2}{x_2-2x_1+x_0}=\tilde{x}_0$$

我们可以证明，按照以下迭代格式修正后得到的 $\{\tilde{x}_k\}_{k=0}^{+\infty}$ 更快地收敛到 α。

$$\tilde{x}_{k+1}=x_k-\frac{(x_{k+1}-x_k)^2}{x_{k+2}-2x_{k+1}+x_k}$$

将以上加速思想与不动点迭代结合，我们定义了一个新的迭代函数

$$\psi(x)=x-\frac{(\varphi(x)-x)^2}{\varphi(\varphi(x))-2\varphi(x)+x}$$

我们可以证明，当 $\varphi'(\alpha)\neq1$ 时，Steffensen 加速法至少是二阶收敛的（即使由 φ 决定的不动点迭代本身不收敛）。

1.3 Newton 迭代法

牛顿迭代法的基本思想是：在 x_k 处对 $f(x)$ 作线性逼近，即过 $(x_k, f(x_k))$ 作 $f(x)$ 的切线，该切线与 x 轴的交点即为 x_{k+1}；也可看成将 $f(x)$ 在 x_k 附近作一阶 Taylor 展开：$f(x) \approx f(x_k) + f'(x_k)(x - x_k) \approx 0$ 得到迭代格式

$$x_{k+1} = x_k - [f'(x_k)]^{-1} f(x_k)$$

我们可以证明，如果 $f \in C[a,b]$ 在 $[a,b]$ 上有零点，且满足 $f''(x)$ 在 $[a,b]$ 上不变号，$f'(x) \neq 0, \forall x \in [a,b]$，$\frac{|f(c)|}{b-a} \leqslant |f'(c)|$，其中 $c \in a, b$ 使得 $|f'(c)| = \min\{|f'(a)|, |f'(b)|\}$，则无论以 $[a,b]$ 中的哪个点为初值，Newton 迭代总是而二阶收敛到 $f(x) = 0$ 的根 α 的。

2 实验设计

由题，现有 $f(x) = x^3 + 2x^2 + 10x - 20$，我们尝试用两种不动点迭代方法，他们的 Steffensen 加速版本，和 Newton 迭代法来求解该方程的根。

2.1 不动点迭代

我们构造了两种不动点迭代格式：

$$x_{k+1} = \frac{20 - 2x_k^2 - x_k^3}{10}; \qquad x_{k+1} = \sqrt[3]{20 - 10x_k - 2x_k^2}$$

从而有以下两个迭代函数 $\varphi_1(x)$ 和 $\varphi_2(x)$：

$$\begin{aligned} &\text{function}[y] = \text{fun1}(x) \\ &y = (20 - 2 * (x^2) - x^3)/10; \\ &\text{end} \end{aligned} \qquad \begin{aligned} &\text{function}[y] = \text{fun2}(x) \\ &y = \text{nthroot}(20 - 10 * x - 2 * x^2, 3); \\ &\text{end} \end{aligned}$$

调用这两个迭代函数，以 x_0 为初始迭代点，并规定迭代收敛的精度 tol，即可得到两种方程的 solver：

```
function [x] = s1(x0,tol)
MAX_ITER = 1000;
x = x0;

for k = 1:MAX_ITER
    funcx = fun1(x);
    err = abs(x - funcx);
    x = funcx;

    if err < tol
        break;
    end
end

if k == MAX_ITER
    sprintf('Algorithm does not converge.');
end
x = vpa(x,8);
end
```

```
function [x] = s2(x0,tol)
MAX_ITER = 1000;
x = x0;

for k = 1:MAX_ITER
    funcx = fun2(x);
    err = abs(x - funcx);
    x = funcx;

    if err < tol
        break;
    end
end

if k == MAX_ITER
    sprintf('Algorithm does not converge.');
end
x = vpa(x,8);
end
```

2.2 Steffensen 加速

由以上实验原理，Steffensen 加速是定义了一个新的迭代函数：

$$\psi(x) = x - \frac{(\varphi(x) - x)^2}{\varphi(\varphi(x)) - 2\varphi(x) + x}$$

由此我们对上面的两个 solver 进行改进得到：

```
function [x] = s1_steff(x0,tol)
MAX_ITER = 1000;
x = x0;

for k = 1:MAX_ITER
    y = fun1(x);
    z = fun1(y);
    funcx = x - (y-x)^2 / (z - 2*y + x);

    err = abs(x - funcx);
    x = funcx;

    if err < tol
        break;
    end
end

if k == MAX_ITER
    sprintf('Algorithm does not converge.');
end
x = vpa(x,8);
end
```

```
function [x] = s2_steff(x0,tol)
MAX_ITER = 1000;
x = x0;

for k = 1:MAX_ITER
    y = fun2(x);
    z = fun2(y);
    funcx = x - (y-x)^2 / (z - 2*y + x);

    err = abs(x - funcx);
    x = funcx;

    if err < tol
        break;
    end
end

if k == MAX_ITER
    sprintf('Algorithm does not converge.');
end
x = vpa(x,8);
end
```

2.3 Newton 迭代

Newton 迭代需要用到 $f(x)$ 与 $f'(x)$，故先构造相应的函数：

function[y] = fun3(x)

$y = x^3 + 2 * x^2 + 10 * x - 20;$

end

function[y] = dfun3(x)

$y = 3 * x^2 + 4 * x + 10;$

end

然后调用这两个函数，以 x_0 为初始迭代点，并规定迭代收敛的精度 tol，得到 Newton 迭代法的 solver：

```
function [x] = Newton(x0,tol)
MAX_ITER = 1000;
x = x0;

for k = 1:MAX_ITER
    x = x - fun3(x) / dfun3(x);

    if abs(fun3(x)) < tol
        break;
    end
end

if k == MAX_ITER
    sprintf('Algorithm does not converge.');
end
x = vpa(x,8);
end
```

3 结果与分析

我们现在拥有了基于四种不同方法求解 $f(x)=0$ 的函数，统一取初始迭代值 $x_0=1$，迭代收敛精度 $\mathrm{tol}=1e-7$，然后运行每一个函数，我们得到结果：

$x=0.54894648$	第一种不动点迭代
$x=-3.1622777$	第二种不动点迭代
$x=1.3688081$	第一种不动点迭代 +Steffensen 加速
$x=1.3688081$	第二种不动点迭代 +Steffensen 加速
$x=1.3688081$	Newton 迭代法

已知精确解 $x^*=1.368808107\ldots$，可见前两种不动点迭代法是不收敛的，之后经过 Steffensen 加速后都收敛了，最后的 Newton 迭代法也是收敛的。

3.1 不动点迭代收敛检验

对于前两种不动点迭代的迭代函数：

$$\varphi_1(x)=\frac{20-2x^2-x^3}{10};\qquad \varphi_2(x)=\sqrt[3]{20-10x-2x^2}$$

我们有

$$\varphi_1'(x)=-\frac{2}{5}x-\frac{3}{10}x^2;\qquad \varphi_1'(\alpha)\approx -1.1096$$

$$\varphi_2'(x)=\frac{-10-4x}{3(20-10x-2x^2)^{2/3}};\qquad \varphi_2'(\alpha)\approx -9.6650$$

即都不满足 $|\varphi'(\alpha)|<1$ 的条件，故都不收敛。

3.2 Steffensen 加速法收敛检验

又由 $\varphi_1'(\alpha)\neq 1,\varphi_2'(\alpha)\neq 1$ 知，Steffensen 加速至少是二阶收敛的。

3.3 牛顿迭代法收敛检验

由题，$f(x)=x^3+2x^2+10x-20, f(1)=-7, f(2)=16$，故在 $[1,2]$ 上函数有零点。$f'(x)=3x^2+4x+10\neq 0,\forall x\in[1,2]$，且 $f''(x)=6x+4>0,\forall x\in[1,2]$。$f'(1)<f'(2)\to c=1, \frac{|f(c)|}{2-1}=|f(1)|=7\leqslant|f'(1)|=17$。由此可知以 $x_0=1\in[1,2]$ 为初始迭代点的 Newton 迭代法收敛。