《计算流体力学基础》第四次作业

姓名：杨阳
学号：150*******
专业：流体力学

采用不同的加权余量法计算下面的常微分方程边值问题：

$$L(u)=\frac{d^2u}{dx^2}+u+x=0,\quad (0\le x\le 1)$$
$$u(0)=u(1)=0$$

近似解取为：

$$u=x(1-x)\left(a_0+a_1x+a_2x^2+a_3x^3+a_4x^4+\ldots+a_nx^n\right)$$

1. 采用手工计算的方法计算
   出于简单的考虑手工计算的时候近似解只考虑前两项：

$$u=x(1-x)(a_0+a_1x)$$

（1）配置法选配置点：$x_1=\frac{1}{3},\quad x_2=\frac{2}{3}$

（2）子区域法取得两个子区域为 $D_1\ :0<x<\frac{1}{2};D_2\ :0<x<1$。

（3）比较配置法，子区域法，最小二乘法，矩法，Galerkin 法的精度。

2. 对任意的 n 编写 **Galerkin** 方法的计算程序，并比较在 n 逐渐增加的时候解得精度改善情况。

问题的求解过程：

考虑到问题中具有大量的符号计算，采用计算机代数系统 Maple 编写计算程序并使用 MATLAB 绘制出计算结果

对于这个边值问题，具有如下形式的解析解：

$$u(x)=\frac{\sin(x)}{\sin(1)}-x$$

下面的近似结果将和这个精确解进行比较。

1. 手工计算

取近似解为：

$$u=x(1-x)(a_0+a_1x)$$

（1） 采用配置法

取近似解：

$$u=x(1-x)(a_0+a_1x)$$

在配置法中的权函数为狄拉克函数，其表达式如下：

$$\int_\Omega (Lu-f)\omega_i d\Omega=0$$

$$Weight:\omega_i=\delta(x-x_i)$$

通过计算可以得到：

$$u=x(1-x)(a_0+a_1x)$$

$$(Lu-f)=-4a_1x-2a_0+2(1-x)a_1+x(1-x)(a_1x+a_0)+x$$

从而得到加权余量法的积分：

$$\int_\Omega (Lu-f)\omega_i d\Omega$$

$$=\int_0^1\left[-4a_1x-2a_0+2(1-x)a_1+x(1-x)(a_1x+a_0)+x\right]\delta(x-x_i)dx$$

令这个积分等于零：

$$\int_\Omega (Lu-f)\omega_i d\Omega=0$$

得到两个线性代数方程组：

$$\left[-4a_1x-2a_0+2(1-x)a_1+x(1-x)(a_1x+a_0)+x\right]\Big|_{x=x_1}=0$$

$$\left[-4a_1x-2a_0+2(1-x)a_1+x(1-x)(a_1x+a_0)+x\right]\Big|_{x=x_2}=0$$

带入具体的数值后得到：

$$\frac{2}{27}a_1-\frac{16}{9}a_0+\frac{1}{3}=0$$

$$-\frac{50}{27}a_1-\frac{16}{9}a_0+\frac{2}{3}=0$$

求解这个线性代数方程组，从而得到近似解中的系数：

$$a_0=\frac{81}{416},\quad a_1=\frac{9}{52}$$

近似解具有如下的表达式：

$$u=x(1-x)\left(\frac{81}{416}+\frac{9}{52}x\right)$$

（2）采用子区域法

子区域的划分方法如下：

$$D_1:0<x<\frac{1}{2};D_2:0<x<1$$

子区域的加权余量法：

$$\int_{\Omega}\left(Lu-f\right)\omega_i d\Omega=0$$

$$Weight:\omega_i=\begin{cases}1, & x\in D_i\\ 0, & x\notin D_i\end{cases}$$

具体的表达式：

$$\begin{aligned}&\int_{\Omega}\left(Lu-f\right)\omega_i d\Omega\\&=\int_0^1\left[-4a_1x-2a_0+2\left(1-x\right)a_1+x\left(1-x\right)\left(a_1x+a_0\right)+x\right]\omega_i dx\end{aligned}$$

令加权余量为零。

$$\int_{\Omega}\left(Lu-f\right)\omega_i d\Omega=0$$

从而得到线性代数方程组：

$$\int_0^{\frac{1}{2}}\left[-4a_1x-2a_0+2\left(1-x\right)a_1+x\left(1-x\right)\left(a_1x+a_0\right)+x\right]\omega_i dx=0$$

$$\int_0^1\left[-4a_1x-2a_0+2\left(1-x\right)a_1+x\left(1-x\right)\left(a_1x+a_0\right)+x\right]\omega_i dx=0$$

将权函数带入上面的积分表达式并进行计算可以得到下面的线性代数方程组。

$$\frac{53}{192}a_1-\frac{11}{12}a_0+\frac{1}{8}=0$$

$$-\frac{11}{12}a_1-\frac{11}{6}a_0+\frac{1}{2}=0$$

求解这一组线性代数方程组得到近似解得系数：

$$a_0=\frac{97}{517},a_1=\frac{8}{47}$$

从而得到具有下面形式的近似解：

$$u=x\left(1-x\right)\left(\frac{97}{517}+\frac{8}{47}x\right)$$

（3）采用矩法

在矩法中权重函数定义为下面的形式：

$$\omega_i = |\mathbf{r}|^{i-1}$$

本问题属于一位问题，具有下列形式的权函数：

$$\omega_1(x)=1, \omega_2(x)=x$$

因此根据加权余量为零

$$\int_\Omega (Lu-f)\omega_i d\Omega = 0$$

可以得到下面的积分表达式

$$\int_0^{\frac{1}{2}} \left[-4a_1 x - 2a_0 + 2(1-x)a_1 + x(1-x)(a_1 x + a_0) + x\right] dx = 0$$

$$\int_0^1 \left[-4a_1 x - 2a_0 + 2(1-x)a_1 + x(1-x)(a_1 x + a_0) + x\right] x dx = 0$$

求出这个积分的具体表达式得到下面的线性代数方程组：

$$-\frac{11}{12}a_1 - \frac{11}{6}a_0 + \frac{1}{2} = 0$$

$$-\frac{19}{20}a_1 - \frac{11}{12}a_0 + \frac{1}{3} = 0$$

求解这一组线性代数方程组可以得到近似解中的系数：

$$a_0 = \frac{122}{649}, \quad a_1 = \frac{10}{59}$$

从而求出具有下面形式的近似解：

$$u(x) = x(1-x)\left(\frac{122}{649}x + \frac{10}{59}\right)$$

（4）最小二乘法

在最小二乘法中目标泛函为：

$$J = \int_\Omega R^2 d\Omega$$

对待定参数求偏导数：

$$\frac{\partial J}{\partial a_i} = \frac{\partial}{\partial a_i}\int_\Omega R^2 d\Omega = \int_\Omega \frac{\partial}{\partial a_i} R^2 d\Omega = \int_\Omega 2R\frac{\partial R}{\partial a_i} d\Omega$$

目标泛函取得驻值的条件：

$$\frac{\partial J}{\partial a_i} = 0$$

具体表达形式：

$$\int_\Omega 2R\frac{\partial R}{\partial a_i} d\Omega = 0$$

因此我们知道在最小二乘加权余量法中的权函数如下：

$$\omega_i = \frac{\partial R}{\partial a_i}$$

对于本问题权函数的具体形式：

$$\omega_0 = -2 + x(1-x), \omega_1 = -6x + 2 + x^2(1-x)$$

令加权余量为零：

$$\int_\Omega (Lu - f)\omega_i d\Omega = 0$$

从而得到一组积分：

$$\int_0^{\frac{1}{2}} \left[-4a_1 x - 2a_0 + 2(1-x)a_1 + x(1-x)(a_1 x + a_0) + x\right]\omega_0 dx = 0$$

$$\int_0^1 \left[-4a_1 x - 2a_0 + 2(1-x)a_1 + x(1-x)(a_1 x + a_0) + x\right]\omega_1 dx = 0$$

求出这两个积分得到线性代数方程组：

$$\frac{101}{60}a_1 + \frac{101}{30}a_0 - \frac{11}{12} = 0$$

$$\frac{131}{35}a_1 + \frac{101}{60}a_0 - \frac{19}{20} = 0$$

求解这个线性代数方程组可以得到近似解中的系数。

$$a_0 = \frac{46161}{246137}, a_1 = \frac{413}{2437}$$

从而知道近似解得形式如下：

$$u = x(1-x)\left(\frac{46161}{246137} + \frac{413}{2437}x\right)$$

（5）Galerkin 法

近似函数具有下面的表达式：

$$u = x(1-x)(a_0 + a_1 x)$$

将其按照待定系数拆分开：

$$u = a_0 x(1-x) + a_1 x^2(1-x)$$

从而可以看出基函数的形式如下：

$$\varphi_0 = x(1-x), \varphi_1 = x^2(1-x)$$

加权余量法的表达式：

$$\int_{\Omega}\left(Lu-f\right)\omega_i d\Omega=0$$

$$Wieght:\omega_i=\varphi_i$$

得到下面的两个积分表达式：

$$\int_0^{\frac{1}{2}}\left[-4a_1x-2a_0+2(1-x)a_1+x(1-x)(a_1x+a_0)+x\right]\omega_0 dx=0$$

$$\int_0^1\left[-4a_1x-2a_0+2(1-x)a_1+x(1-x)(a_1x+a_0)+x\right]\omega_1 dx=0$$

将权函数的具体表达式带入并积分可以得到下面两个关于近似解中待定系数的线性代数方程组：

$$-\frac{3}{20}a_1-\frac{3}{10}a_0+\frac{1}{12}=0$$

$$-\frac{13}{105}a_1-\frac{3}{20}a_0+\frac{1}{20}=0$$

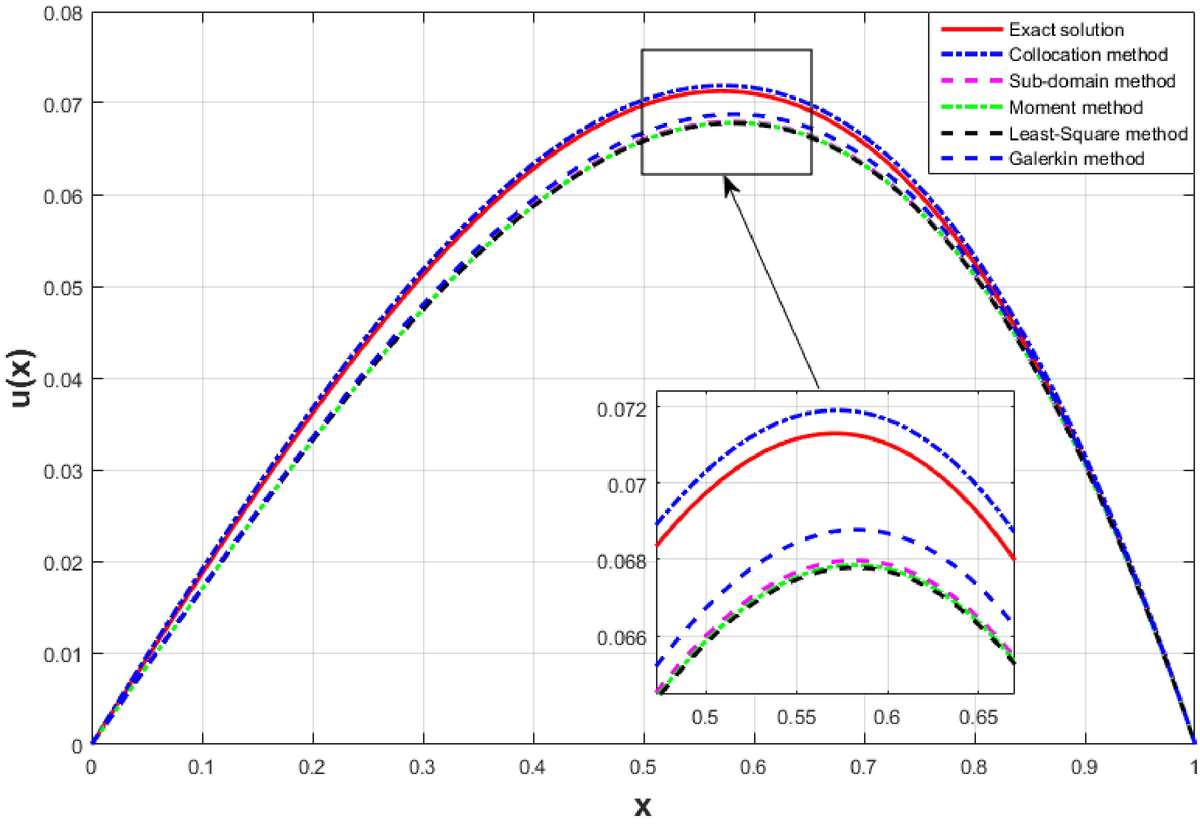
求解这两个待定参数可以得到：

$$a_0=\frac{71}{369},a_1=\frac{7}{41}$$

从而知道近似解具有下面的形式：

$$u=x(1-x)\left(\frac{71}{369}x+\frac{7}{41}\right)$$

下面画出这些不同方法解得曲线图进行对比：



分析这个结果可以出配点法的结果离精确解最近，而其次是伽辽金法离精确解的结果较近，其余的子区域法、矩法、最小二乘法的结果也比较靠近。理论上最小二乘加权余量法和伽辽金加权余量法的精度较高，但是因为 n 比较小加上可能是配置法中点的取得比较合使得配点法的结果较好。而且可以发现解得偏差主要是出现在中心区域。

2. 对任意的 n 编写 Galerkin 方法的计算程序，并比较在 n 逐渐增加的时候解得精度改善情况。

任意 n 时的近似函数：

$$u = x(1-x)\left(a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 \% x^4 + \ldots + a_n x^n\right)$$

从而可以知道基函数具有下面的形式：

$$\begin{aligned}
&\varphi_0(x) = x(1-x) \\
&\varphi_1(x) = x^2(1-x) \\
&\varphi_2(x) = x^3(1-x) \\
&\ldots \\
&\varphi_n(x) = x^{n+1}(1-x)
\end{aligned}$$

从而近似解可以表示成如下形式：

$$u(x) = \sum_{i=0}^{n} a_i \varphi_i(x)$$

在 Galerkin 方法中权函数就是基函数：

$$\omega_i(x) = \varphi_i(x)$$

加权余量法的表达式：

$$\begin{aligned}
&\int_\Omega (Lu - f)\omega_i d\Omega = 0 \\
&Wieght : \omega_i = \varphi_i
\end{aligned}$$

对于这个问题的具体形式是：

$$\int_0^1 (Lu - f)\omega_i(x) dx = 0$$

求出上面这个积分表达式从而可以得到 n 个线性代数方程。

$$\begin{aligned}
&f_1(a_0, a_1, \ldots, a_n) = 0 \\
&f_2(a_0, a_1, \ldots, a_n) = 0 \\
&f_3(a_0, a_1, \ldots, a_n) = 0 \\
&\ldots \\
&f_n(a_0, a_1, \ldots, a_n) = 0
\end{aligned}$$

求解这组线性代数方程组可以得到近似解中的待定参数：

$$a_0, a_1, \ldots, a_n$$

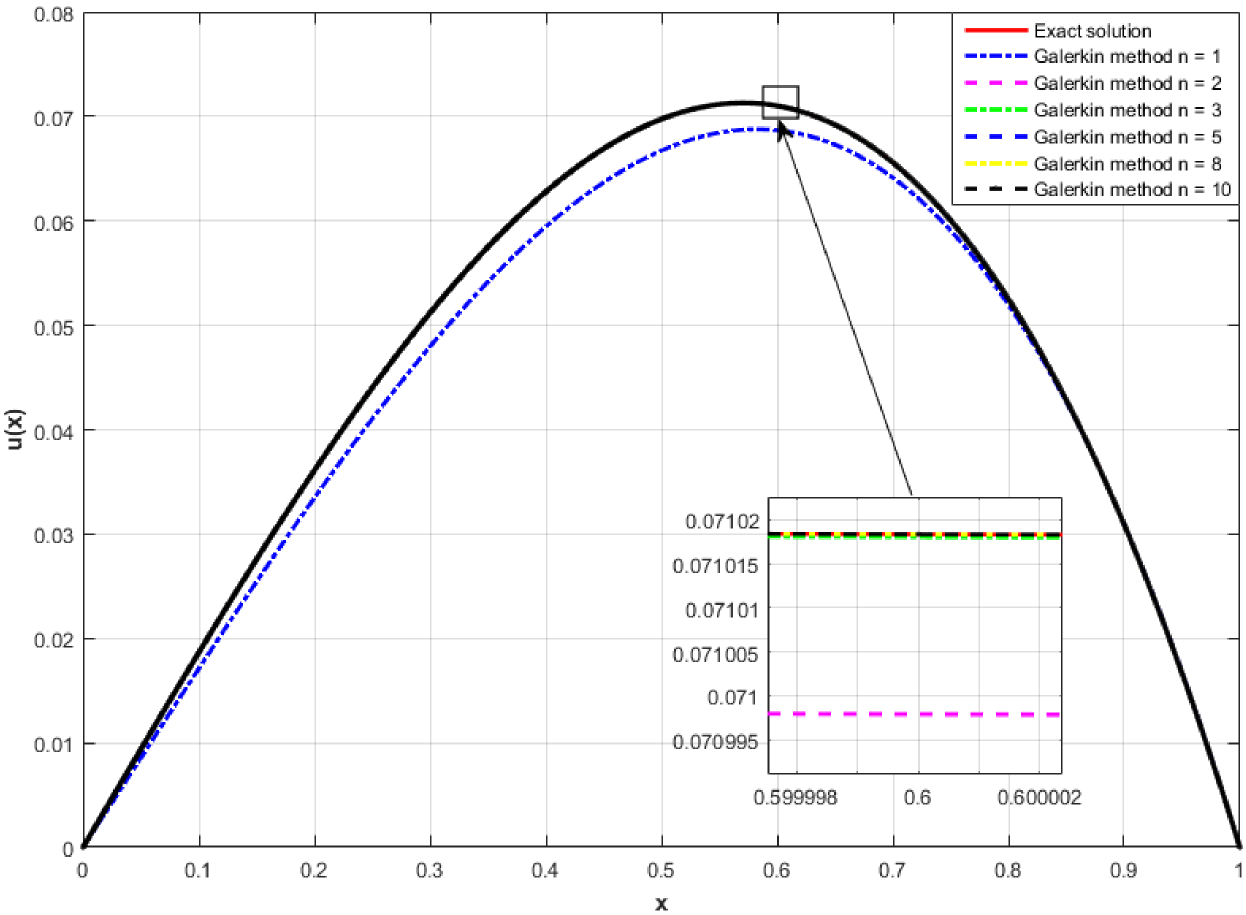
从而可以知道近似解得表达式：

$$u = x(1-x)\left(a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + \ldots + a_n x^n\right)$$

具体编程的问题的分析：

我们的目的是求出近似表达式中的待定系数，为 Galerkin 法最后会得到关于这些系数的线性代数方程组，在这些方程组中最重要的就是这些变量前面的系数和常数项的具体数值，而这些数值与近似解的阶数 n 有关，获得这些系数可以进行详细分析后编写出求出这些系数的算法，这样对于较大的 n 仍能具有较快的计算速度。而在这里求解过程中借助计算机代数系统 Maple 进行符号求解，得到这些系数并求解线性代数方程组。

Galerkin 法的解随 n 的变化的变化：



从结果中可以看出在 n=1 的时候与精确解得偏差较大，当 n=2 以后偏差迅速减小，n=3 以上解与精确解的偏差基本上很难分辨，这也提示我们不用为了追求单方面的精度而过度提高阶数 n，反而会带来巨大的计算量，当 n=10 的时候多项式的项数已经相当多了。

程序附录

1. 任意阶 n 的 Galerkin 方法计算 Maple 程序

Galerkin.mw

```
>
```

Exact solution:

```
> eq:= diff(u(x), [`$`(x, 2)])+u(x)+x = 0;
```

$$eq := \frac{d^2}{dx^2} u(x) + u(x) + x = 0$$

```
> ics := u(0)=0, u(1)=0;
```

$$ics := u(0) = 0,\ u(1) = 0$$

```
> exact_u:=rhs(dsolve({eq, ics}));
```

$$exact\_u := \frac{\sin(x)}{\sin(1)} - x$$

```
> CodeGeneration['Matlab']( '' );
cg7 = sin(x) / sin(0.1e1) - x;
> p1 := plot(exact_u,x=0..1,color='blue'):
>
```

Galerkin method:

```
>
```

Define order:

```
> n:=10;
```

$$n := 10$$

Consturct test function:

```
> u_test:=0;
```

$$u\_test := 0$$

```
> for i from 0 to n do
    phi||i := x^(i+1)*(1-x);
    u_test := u_test + a||i*phi||i;
end:
>
```

Compute Operator:

```
> Lu:=diff(u_test, [`$`(x, 2)])+ u_test + x:
```

Construct equations:

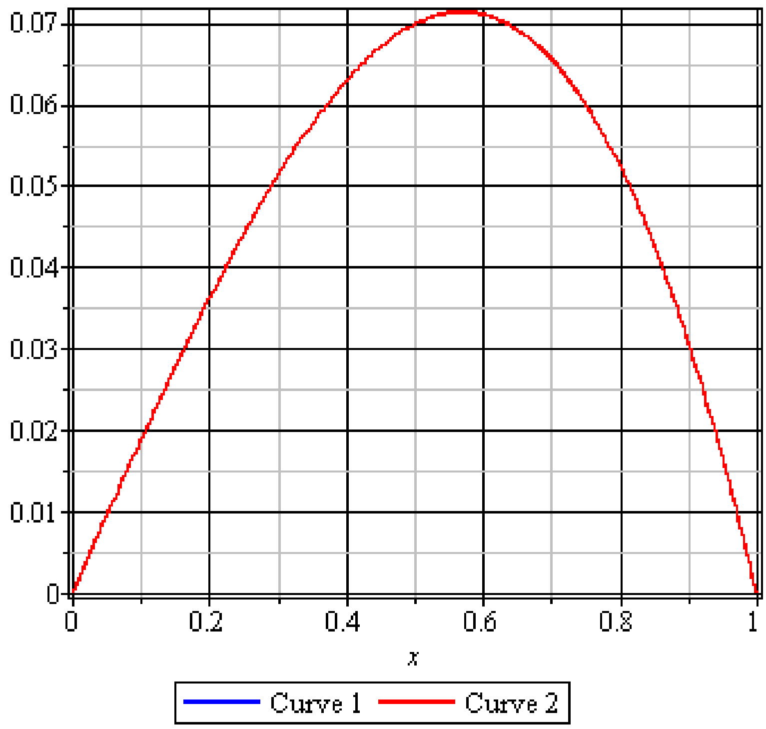
```
> for i from 0 to n do
     eq||i := int(Lu*phi||i,x=0..1);
end:
>
```

Solve equations:

```
> with(LinearAlgebra):
> nn := n + 1:
> A:=Matrix(nn,nn,0):
> b:=Vector(nn,0):
> for i from 1 to nn do
    ii := i - 1:
    b[i] := eq||ii:
    for j from 1 to nn do
         jj := j - 1:
        A[i,j] := coeff(eq||ii,a||jj):
        b[i] := b[i] - A[i,j] * a||jj:
    end:
    b[i] := -b[i]:
end:
>
> a_result:=LinearSolve(A,b):
>
>
```

Solution:

```
> u_app:=0:
> for i from 0 to n do
    u_app := u_app + a_result[i+1]*phi||i;
end:
>
> p2:=plot(u_app,x=0..1,color='red'):
> with(plots):
> display(p1,p2);
```




```
>
>
> u_app;
```

$$\frac{4733180254110884058499825 75}{2512369010098021442164037778}\ x\ (1-x)$$

$$+\frac{788863375685412223209895 50}{418728168349670240360672963}\ x^{2}\ (1-x)$$

$$-\frac{404941341850424444547600 0}{418728168349670240360672963}\ x^{3}\ (1-x)$$

$$-\frac{404941340996846780875537 0}{418728168349670240360672963}\ x^{4}\ (1-x)$$

$$+\frac{9737407771953737130731 0}{418728168349670240360672963}\ x^{5}\ (1-x)$$

$$+\frac{9737435806080067643200 2}{418728168349670240360672963}\ x^{6}\ (1-x)$$

$$-\frac{1359523407693545994408}{418728168349670240360672963}\ x^{7}\ (1-x)$$

$$-\frac{1357794784581134597558}{418728168349670240360672963}\ x^{8}\ (1-x)$$

$$+\frac{11079298114176799757}{418728168349670240360672963}\ x^{9}\ (1-x)$$

$$+\frac{40076222660103319736}{1256184505049010721082018889}\ x^{10}\ (1-x)$$

$$-\frac{371450}{313853699470081}\ x^{11}\ (1-x)$$

```
> CodeGeneration['Matlab'](u_app);
cg8 = 0.4733180254110884058499825 75e27 /
0.2512369010098021442164037778e28 * x * (1 - x) +
0.788863375685412223209895 50e26 / 0.418728168349670240360672963e27
* (x ^ 2) * (1 - x) - 0.404941341850424444547600 0e25 /
0.418728168349670240360672963e27 * (x ^ 3) * (1 - x) -
0.404941340996846780875537 0e25 / 0.418728168349670240360672963e27 *
(x ^ 4) * (1 - x) + 0.9737407771953737130731 0e23 /
0.418728168349670240360672963e27 * (x ^ 5) * (1 - x) +
0.9737435806080067643200 2e23 / 0.418728168349670240360672963e27 *
(x ^ 6) * (1 - x) - 0.1359523407693545994408e22 /
0.418728168349670240360672963e27 * (x ^ 7) * (1 - x) -
0.1357794784581134597558e22 / 0.418728168349670240360672963e27 * (x
^ 8) * (1 - x) + 0.11079298114176799757e20 /
0.418728168349670240360672963e27 * (x ^ 9) * (1 - x) +
0.40076222660103319736e20 / 0.1256184505049010721082018889e28 * (x
```

```
^ 10) * (1 - x) - 0.371450e6 / 0.313853699470081e15 * (x ^ 11) * (1
- x);
>
```

2. 画图比较计算结果的 MATLAB 程序

ComparePlot.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%
%%%% Compare different numerical method in
Weight residual method
%%%% ComparePlot.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%
%% define silutions
x =
0:0.001:1;
                     % x span
u = sin(x) / sin(0.1e1) -
x;
% exact solution
u1 = x .* (1 - x) .* (0.9e1 / 0.52e2 .* x
+ 0.81e2 / 0.416e3);           %
Collocation method
```

```
u2 = x .* (1 - x) .* (0.97e2 / 0.517e3 .*
x + 0.8e1 / 0.47e2);              % Sub-
domain method
u3 = x .* (1 - x) .* (0.122e3 / 0.649e3 .*
x + 0.10e2 / 0.59e2);            % Moment
method
u4 = x .* (1 - x) .* (0.46161e5 /
0.246137e6 .* x + 0.413e3 / 0.2437e4);   %
Least-Square method
%%%% Galerkin method
u5_1 = x .* (1 - x) .* (0.71e2 /
0.369e3 .* x + 0.7e1 /
0.41e2);                    % Galerkin method n =
1
u5_2 = 0.13811e5 / 0.73554e5 .* x .* (1 -
x) ...
    + 0.2380e4 / 0.12259e5 .* (x .^ 2) .*
(1 - x) ...
    - 0.7e1 ./ 0.299e3 .* (x .^ 3) .* (1 -
x);                                  % Galerkin
method n = 2
u5_3 = 0.1297898e7 / 0.6889857e7 .* x .*
```

```
(1 - x) ...
    + 0.433198e6 / 0.2296619e7 .* (x .^
2) .* (1 - x) ...
    - 0.24166e5 / 0.2296619e7 .* (x .^
3) .* (1 - x) ...
    - 0.66e2 / 0.7681e4 .* (x .^ 4) .* (1
- x);                               % Galerkin
method n = 3
u5_5 = 0.4855505939e10 /
0.25772989853e11 .* x .* (1 - x) ...
    + 0.179652917103e12 /
0.953600624561e12 .* (x .^ 2) .* (1 -
x) ...
    - 0.9215765647e10 /
0.953600624561e12 .* (x .^ 3) .* (1 -
x) ...
    - 0.711074859e9 / 0.73353894197e11 .*
(x .^ 4) .* (1 - x) ...
    + 0.19857871e8 / 0.73353894197e11 .*
(x .^ 5) .* (1 - x) ...
    + 0.715e3 / 0.3484249e7 .* (x .^ 6) .*
(1 - x);                            % Galerkin
```

```
method n = 5
u5_8 = 0.20871747307129307769 7e21 /
0.11078709938217386953 78e22 .* x .* (1 -
x) ...
    + 0.10435873648782963968 6e21 /
0.55393549691086934768 9e21 .* (x .^ 2) .*
(1 - x) ...
    - 0.53569682231892419 71e19 /
0.55393549691086934768 9e21 .* (x .^ 3) .*
(1 - x) ...
    - 0.53569757312176026 22e19 /
0.55393549691086934768 9e21 .* (x .^ 4) .*
(1 - x) ...
    + 0.12884649721645761 8e18 /
0.55393549691086934768 9e21 .* (x .^ 5) .*
(1 - x) ...
    + 0.12873503275906059 8e18 /
0.55393549691086934768 9e21 .* (x .^ 6) .*
(1 - x) ...
    - 0.16650293828287 60e16 /
0.55393549691086934768 9e21 .* (x .^ 7) .*
(1 - x) ...
```

```
    - 0.1927385453158594e16 /
0.553935496910869347689e21 .* (x .^ 8) .*
(1 - x) ...
    + 0.29393e5 / 0.188326235369e12 .*
(x .^ 9) .* (1 - x);                %
Galerkin method n = 8
u5_10 = 0.473318025411088405849982575e27 /
0.2512369010098021442164037778e28 .* x .*
(1 - x) ...
    + 0.78886337568541222320989550e26 /
0.418728168349670240360672963e27 .* (x .^
2) .* (1 - x) ...
    - 0.4049413418504244445476000e25 /
0.418728168349670240360672963e27 .* (x .^
3) .* (1 - x) ...
    - 0.4049413409968467808755370e25 /
0.418728168349670240360672963e27 .* (x .^
4) .* (1 - x) ...
    + 0.97374077719537371307310e23 /
0.418728168349670240360672963e27 .* (x .^
5) .* (1 - x) ...
    + 0.97374358060800676432002e23 /
```

```
0.418728168349670240360672963e27 .* (x .^
6) .* (1 - x) ...
    - 0.1359523407693545994408e22 /
0.418728168349670240360672963e27 .* (x .^
7) .* (1 - x) ...
    - 0.1357794784581134597558e22 /
0.418728168349670240360672963e27 .* (x .^
8) .* (1 - x) ...
    + 0.1107929811417679975 7e20 /
0.418728168349670240360672963e27 .* (x .^
9) .* (1 - x) ...
    + 0.4007622266010331973 6e20 /
0.1256184505049010721082018889e28 .* (x .^
10) .* (1 - x) ...
    - 0.371450e6 / 0.313853699470081e15 .*
(x .^ 11) .* (1 - x);          %
Galerkin method n = 10

%% plot
figure('Color',[1 1 1]);
plot(x,u,'r','linewidth',2);
hold on;
```

```
plot(x,u1,'b-.','linewidth',2);
plot(x,u2,'m--','linewidth',2);
plot(x,u3,'g-.','linewidth',2);
plot(x,u4,'k--','linewidth',2);
plot(x,u5_1,'b--','linewidth',2);
hold off;
legend('Exact solution','Collocation
method',...
    'Sub-domain method','Moment
method','Least-Square method','Galerkin
method');
grid on;


figure('Color',[1 1 1]);
plot(x,u,'r','linewidth',2);
hold on;
plot(x,u5_1,'b-.','linewidth',2);
plot(x,u5_2,'m--','linewidth',2);
plot(x,u5_3,'g-.','linewidth',2);
plot(x,u5_5,'b--','linewidth',2);
plot(x,u5_8,'y-.','linewidth',2);
plot(x,u5_10,'k--.','linewidth',2);
```

```
hold off;

legend('Exact solution','Galerkin method n
= 1','Galerkin method n = 2',...
    'Galerkin method n = 3','Galerkin
method n = 5',...
    'Galerkin method n = 8','Galerkin
method n = 10');

grid on;

%% Program end
```