Gaussian quadrature

To write a Matlab program using Gaussian quadrature (Gauss-Legendre rule), first you need to know the weights c_i and nodes x_i. A typical table of Gauss-Legendre rule looks like the following:

n (# of points)	x_i	c_i
2	0.5773502691896257	1.0000000000000000
	-0.5773502691896257	1.0000000000000000
3	0.7745966692414834	0.5555555555555556
	0	0.8888888888888888
	-0.7745966692414834	0.5555555555555556
4	0.8611363115940525	0.3478548451374544
	0.3399810435848563	0.6521451548625460
	-0.3399810435848563	0.6521451548625460
	-0.8611363115940525	0.3478548451374544
...	...	...

Then we can use the formula

$$\int_{-1}^{1} f(x)\,dx \approx \sum_{i=1}^{n} c_i f(x_i),$$

which has the degree of accuracy $2n-1$. In other words, the above formula is exact for any polynomial $f(x)$ with degree up to $2n-1$.

If you need to integrate $f(x)$ on the interval $[a,b]$, simply use a change of variable

$$\int_a^b f(x)\,dx = \int_{-1}^{1} \frac{b-a}{2} f\left(\frac{(b-a)t+(b+a)}{2}\right) dt \approx \sum_{i=1}^{n} c_i \frac{b-a}{2} f\left(\frac{(b-a)x_i+(b+a)}{2}\right)$$

Indeed, we can define

$$\tilde{c}_i = c_i \frac{b-a}{2}, \qquad \tilde{x}_i = \frac{(b-a)x_i+(b+a)}{2},$$

then the formula can be written as

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} \tilde{c}_i f(\tilde{x}_i).$$

Next, let use look at three Matlab examples of using the Gauss-legendre rule.

Example 1 Compute $\int_{-1}^{1} e^x \cos x\,dx$ using a Gaussian quadrature with 3 points. We know that its exact value is

$$\int_{-1}^{1} e^x \cos x\,dx = \left(\frac{1}{2}e^x \cos x + \frac{1}{2}e^x \sin x\right)\Big|_{-1}^{1} = 1.933421497\cdots$$

```
>> format long e
>> x = [0.7745966692414834,  0,  -0.7745966692414834];
>> c = [0.5555555555555556, 0.8888888888888888, 0.5555555555555556];
```

```
>> f = exp(x).*cos(x);
>> value = sum(c.*f)

value =

     1.933390469264298e+00
```

Example 2 Compute $\int_{0.5}^{1.5} e^x \cos x \, dx$ using a Gaussian quadrature with 3 points. We know that its exact value is

$$\int_{0.5}^{1.5} e^x \cos x \, dx = \left(\frac{1}{2} e^x \cos x + \frac{1}{2} e^x \sin x \right) |_{-1}^{1} = 1.275078201 \cdots$$

```
>> x = [0.7745966692414834,  0,  -0.7745966692414834];
>> c = [0.5555555555555556, 0.8888888888888888, 0.5555555555555556];
>> a = 0.5;
>> b = 1.5;
>>
>> tildec = (b-a)/2*c;
>> tildex = (b-a)/2*x + (b+a)/2;
>> f = exp(tildex).*cos(tildex);
>> value = sum(tildec.*f)

value =

     1.275069036575852e+00
```

Example 3 From Example 2, we can see that it is convenient to compute $\tilde{c}_i$ and $\tilde{x}_i$ before we apply the gaussian quadrature. These can be written in a Matlab function. One of such function is available on the Matlab File Exchange Center. Simply go to

`http://www.mathworks.com/matlabcentral/fileexchange/4540`

and download the files. You will have a file named `lgwt.m` under the directory. The function is defined as

```
[x, c] = lgwt(n, a, b)
```

Here n is the number of points, $[a, b]$ is the interval, and the function returns x and c. For example, if you want to know what are the values of x and c for a 2-point formula on $[-1, 1]$, try the following:

```
>> [x, c] = lgwt(2,-1,1)

x =

     5.773502691896257e-01
    -5.773502691896257e-01
```

```
c =

     9.999999999999998e-01
     9.999999999999998e-01
```

For a 3-point formula on $[-1, 1]$,

```
>> [x, c] = lgwt(3,-1,1)

x =

     7.745966692414834e-01
                         0
    -7.745966692414834e-01


c =

     5.555555555555544e-01
     8.888888888888888e-01
     5.555555555555544e-01
```

And if you would like to know a 3-point formula on $[0.5, 1.5]$,

```
>> [x, c] = lgwt(3,0.5,1.5)

x =

     1.387298334620742e+00
     1.000000000000000e+00
     6.127016653792582e-01


c =

     2.777777777777772e-01
     4.444444444444444e-01
     2.777777777777772e-01
```

Now, to comput $\int_{0.5}^{1.5} e^x \cos x\, dx$, you can try the following:

```
>> [x, c] = lgwt(3,0.5,1.5);
>> f = exp(x).*cos(x);
>> value = sum(c.*f)

value =

     1.275069036575850e+00
```