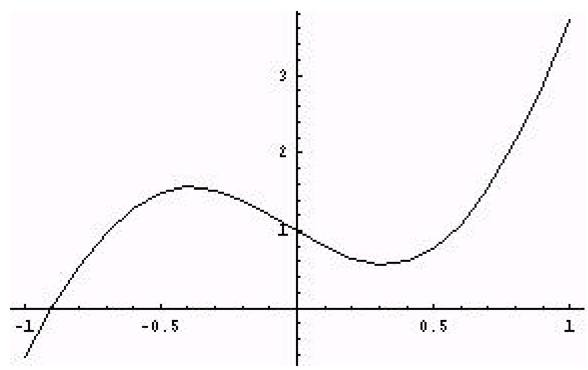
Partial Solutions of Review Problems 2, 1998 Fall

MEAM 501 Analytical Methods in Mechanics and Mechanical Engineering

1. Define the Legendre polynomials in an interval (-1,1), and approximate a data by the least squares method for appropriate number of terms of the basis functions.

-1.	-0.632121
-0.9	-0.0134133
-0.8	0.525114
-0.7	0.962602
-0.6	1.28387
-0.5	1.48153
-0.4	1.55738
-0.3	1.52284
-0.2	1.39852
-0.1	1.21285
0	1.
0.1	0.797154
0.2	0.641618
0.3	0.567842
0.4	0.604768
0.5	0.773721
0.6	1.08706
0.7	1.54774
0.8	2.14976
0.9	2.87959
1.	3.71828



Legendre polynomials are defined as the polynomials obtained by the othogonalization of the polynomial basis functions $\{1 \quad x \quad x^2 \quad x^3 \quad \ldots \quad x^n \quad \ldots\}$ with respect to an inner product (.,.) defined on a given interval, say, (-1,1) or (0,1):

$$(f, g) = \int_{-1}^{1} fg dx .$$

In general, they are also normalized by its natural norm $\|f\| = \sqrt{(f, f)}$.

```
n=6
pbasis=Table[x^(i-1),{i,1,n+1}]
LP=pbasis;
LP[[1]]=pbasis[[1]]/Sqrt[NIntegrate[pbasis[[1]]^2,{x,-1,1}]];
Do[xi=pbasis[[i]];
```

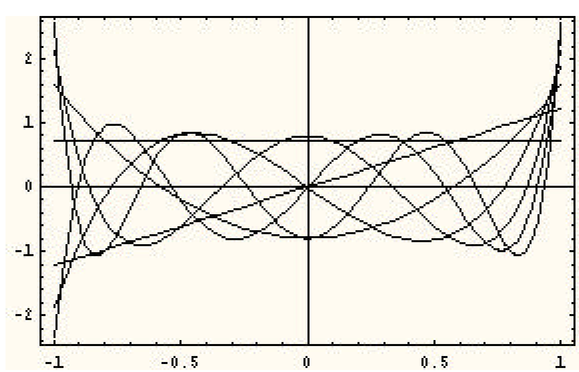
```
    Do[cj=NIntegrate[LP[[j]]*xi,{x,-1,1}];
            xi=xi-cj*LP[[j]],{j,1,i-1}];
            LP[[i]]=Expand[xi/Sqrt[NIntegrate[xi^2,{x,-1,1}]]],
    {i,2,n+1}]
LP
Plot[Release[LP],{x,-1,1},PlotRange->All,Frame->True]
```

Polynomial Basis Functions

```
{1, x, x^2, x^3, x^4, x^5, x^6}
```

Legendre Polynomial Computed

```
{0.7071067811865474 61`,
 0.` + 1.22474487139158894` x,
 -0.790569415042094725`+0. ` x + 2.37170824512628408` x^2,
 0.` - 2.80624304008045655` x + 0.` x^2 + 4.67707173346742699` x^3,
 0.795495128834865994` + 0.` x -
  7.95495128834866083` x^2 + 0.` x^3 + 9.28077650307343837` x^4,
 0.` + 4.39726477483446664` x + 0.` x^2 -
  20.5205689492275089` x^3 + 0.` x^4 + 18.4685120543047567` x^5,
 -0.796721798998875385` + 0.` x + 16.7311577789763576` x^2 +
  0.` x^3 - 50.1934733369290153` x^4 + 0.` x^5 + 36.8085471137479203` x^6}
```



Let a function f be approximated by

$$f(x) \approx f_n(x) = \sum_{i=1}^{n} c_i L_i(x)$$

using the least squares method, that is, the coefficients c_i are determined by solving rectangular equations with the pseudo-inverse, i.e., the singular value decomposition :

$$\sum_{i=1}^{n} c_i L_i(x_m) = f(x_m) \quad , \quad m = 1,2,....,21$$

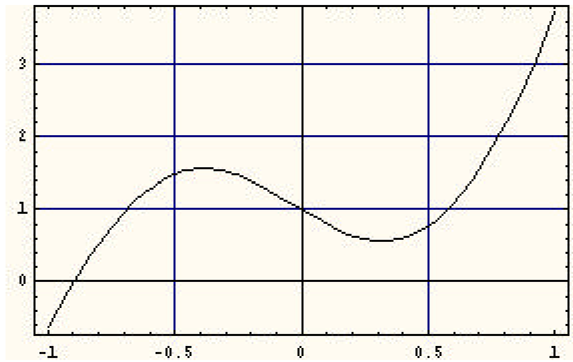
where x_m are the coordinates at which the function f is sampled.

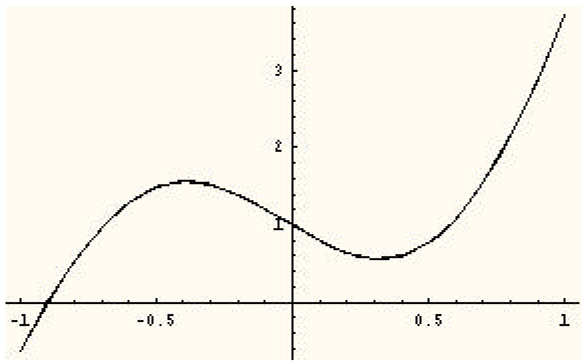
.

```
n=10;
A=N[Table[LegendreP[i-1,x]/.{x->data[[j,1]]},{j,1,21},{i,1,n}]];
coef=PseudoInverse[A].Transpose[data][[2]];
fn=Sum[coef[[i]]*LegendreP[i-1,x],{i,1,n}]
g2=Plot[fn,{x,-1,1},PlotRange->All, Frame->True, GridLines->Automatic]
Show[{g1,g2}]
fnj=Table[N[fn/.{x->data[[j,1]]}],{j,1,21}];
errornorm=Sqrt[(fnj-Transpose[data][[2]]).(fnj-Transpose[data][[2]])]
```

Approximated Function $f_n(x)$

$$\begin{aligned}
&1.1752 + 0.748711\,x + 0.357814\left(-\frac{1}{2} + \frac{3x^2}{2}\right) + 1.6287\left(-\frac{3x}{2} + \frac{5x^3}{2}\right) + \\
&0.00996513\left(\frac{3}{8} - \frac{15x^2}{4} + \frac{35x^4}{8}\right) - 0.218185\left(\frac{15x}{8} - \frac{35x^3}{4} + \frac{63x^5}{8}\right) + \\
&0.0000994548\left(-\frac{5}{16} + \frac{105x^2}{16} - \frac{315x^4}{16} + \frac{231x^6}{16}\right) + \\
&0.0166542\left(-\frac{35x}{16} + \frac{315x^3}{16} - \frac{693x^5}{16} + \frac{429x^7}{16}\right) + \\
&5.06826\times10^{-7}\left(\frac{35}{128} - \frac{315x^2}{32} + \frac{3465x^4}{64} - \frac{3003x^6}{32} + \frac{6435x^8}{128}\right) - \\
&0.000680511\left(\frac{315x}{128} - \frac{1155x^3}{32} + \frac{9009x^5}{64} - \frac{6435x^7}{32} + \frac{12155x^9}{128}\right)
\end{aligned}$$

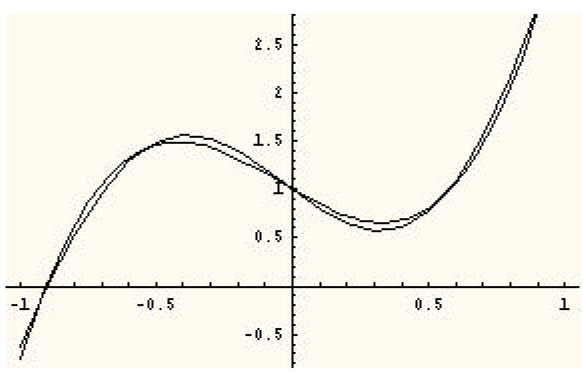




Error for $n = 10$:

0.0000152935

For n = 3, we have large error, say 0.35259.



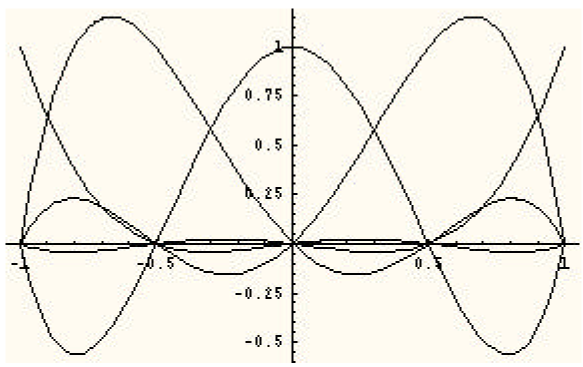
2. Interpolate the above data by using the Lagrange Polynomials by using 3, 5, 11, and 21 basis functions.

Using n+1 points, $x_1, x_2, \ldots\ldots, x_{n+1}$, the n degree Lagrange polynomial is defined by

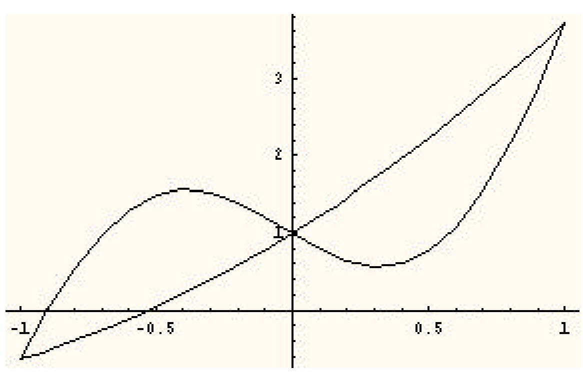
$$L_i(x) = \prod_{\substack{j=1 \\ j \neq i}}^{n+1} \frac{x - x_j}{x_i - x_j} \quad , \quad i = 1,2,\ldots\ldots,n+1 .$$

Lagrange Polynomials (n = 5)

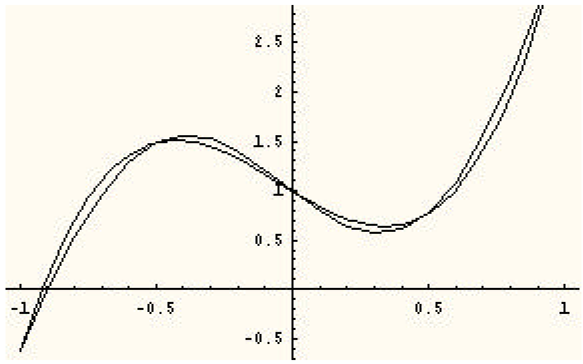
$$\begin{pmatrix} -\frac{2}{3}\,(1-x)\left(-\frac{1}{2}+x\right)x\left(\frac{1}{2}+x\right) \\ \frac{8}{3}\left(\frac{1}{2}-x\right)(-1+x)\,x\,(1+x) \\ -4\,(1-x)\left(-\frac{1}{2}+x\right)\left(\frac{1}{2}+x\right)(1+x) \\ -\frac{8}{3}\,(-1+x)\,x\left(\frac{1}{2}+x\right)(1+x) \\ \frac{2}{3}\left(-\frac{1}{2}+x\right)x\left(\frac{1}{2}+x\right)(1+x) \end{pmatrix}$$



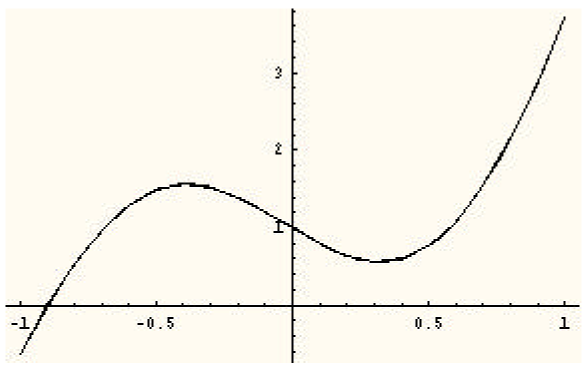
Interpolation/Approximation (n = 3)



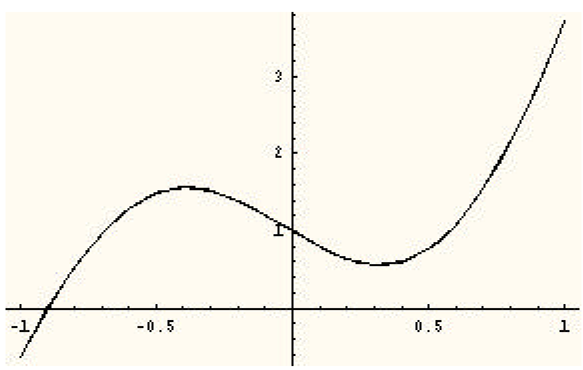
Interpolation/Approximation (n = 5)



Interpolation/Approximation (n = 11)



Interpolation/Approximation (n = 21)



3. *Decomposing the above data into two sets for (-1,0) and (0,+1) for t, approximate the above data by using the Hermite cubic polynomials, Bezier polynomilas, and also B-spline for k=5.*

For the case of the cubic Hermite Polynomial:

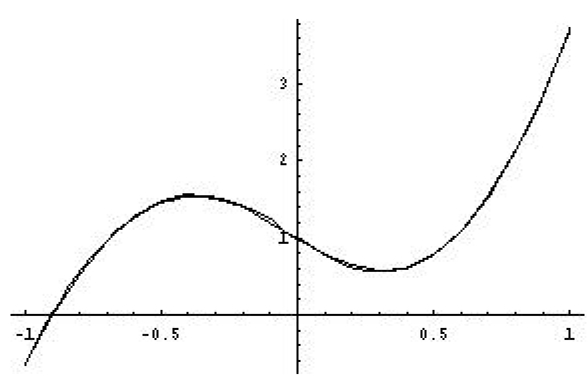
```
HP={1-3*t^2+2*t^3,t*(1-t)^2,3*t^2-2*t^3,(-1+t)*t^2};
n=4;
A1=N[Table[HP[[i]]/.{t->data[[j,1]]+1},{j,1,11},{i,1,n}]];
d1=Table[data[[j,2]],{j,1,11}];
A2=N[Table[HP[[i]]/.{t->data[[10+j,1]]},{j,1,11},{i,1,n}]];
d2=Table[data[[10+j,2]],{j,1,11}];
coef1=PseudoInverse[A1].d1;
coef2=PseudoInverse[A2].d2;
fn1=Sum[coef1[[i]]*(HP[[i]]/.{t->1+x}),{i,1,n}]
fn2=Sum[coef2[[i]]*(HP[[i]]/.{t->x}),{i,1,n}]
g21=Plot[fn1,{x,-1,0},PlotRange->All, Frame->True, GridLines->Automatic]
g22=Plot[fn2,{x,0,1},PlotRange->All, Frame->True, GridLines->Automatic]
Show[{g1,g21,g22}]
```

Subdomain (-1, 0)

$$7.37086\,x^2\,(1+x) - 3.00395\,x\,(1+x)^2 + 0.974231\,(3\,(1+x)^2 - 2\,(1+x)^3) - 0.657877\,(1-3\,(1+x)^2 + 2\,(1+x)^3)$$

Subdomain (0, +1)

$$-2.98088\,(1-x)^2\,x + 9.69654\,(-1+x)\,x^2 + 3.74334\,(3\,x^2 - 2\,x^3) + 1.02509\,(1-3\,x^2+2\,x^3)$$



For Bezier and B-splines, please work out by yourself. However, you should review what is the property of the Bezier and B-splines. Their special characteristics must be well reviewed.

Problem 4, 5, and 6 will be solved by using the Legendre polynomials obtained in Problem1. And the following MATHEMATICA program:

```
n=7;
A=N[Table[LegendreP[i-1,x]/.{x->data[[j,1]]},{j,1,21},{i,1,n}]];
coef=PseudoInverse[A].Transpose[data][[2]];
fn=Sum[coef[[i]]*LegendreP[i-1,x],{i,1,n}]
g2=Plot[fn,{x,-1,1},PlotRange->All, Frame->True, GridLines->Automatic]
Show[{g1,g2}]
fnj=Table[N[fn/.{x->data[[j,1]]}],{j,1,21}];
errornorm=Sqrt[(fnj-Transpose[data][[2]]).(fnj-Transpose[data][[2]])]
dfn=D[fn,x];
Lcurve=NIntegrate[Sqrt[1+dfn^2],{x,-1,1}]
fp={x,fn};
dfp=D[fp,x];
tv=dfp/Sqrt[dfp.dfp];
ftv=Table[N[{fp,tv}/.{x->-1+2*(i-1)/10}],{i,1,11}];
g21=ListPlotVectorField[ftv]
dtv=D[tv,x];
nv=dtv/Sqrt[dtv.dtv];
fnv=Table[N[{fp,nv}/.{x->-1+2*(i-1)/10}],{i,1,11}];
g22=ListPlotVectorField[fnv]
Show[{g2,g21,g22}]
kappa=Sqrt[dtv.dtv]/Sqrt[dfp.dfp];
Plot[kappa,{x,-1,1},PlotRange->All,Frame->True,GridLines->Automatic]
```

4. Compute the total length of the curve defined by the above data.

Let a function $y = f(x)$ be formed from the given data by one of appropriate polynomial form using Lagrange polynomials, Bezier splines, and others. Assuming that the coordinate s is set up along the curve, and let the left end point be zero, while the right end point be set up the total length of the curve L. Then, it can be computed by

$$L = \int ds = \int \sqrt{dx^2 + dy^2} = \int_{-1}^{1} \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx = \int_{-1}^{1} \sqrt{1 + \left(\frac{df}{dx}\right)^2} dx = 6.8478 .$$

5. *Define, compute, and plot the unit tangent and normal vectors of the curve defined by the above data.*

The position vector **r** of an arbitrary point P of the curve is defined by the two-dimensional coordinate $\left(x, f\left(x\right)\right)$, and it can be also defined by the coordinate along the curve s. Then, the tangent vector **t** is defined by

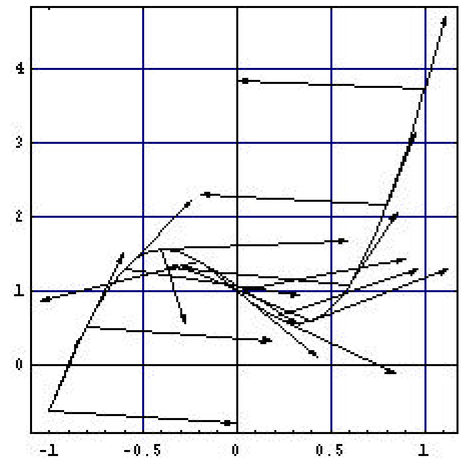
$$\mathbf{t} = \frac{d\mathbf{r}}{ds} = \frac{dx}{ds}\frac{d\mathbf{r}}{dx} = \frac{1}{\sqrt{1 + \left(\frac{df}{dx}\right)^2}} \frac{d\mathbf{r}}{dx} .$$

Since the tangent vector **t** is a unit vector, we also have

$$\frac{d\mathbf{t}}{ds} \bullet \mathbf{t} = 0 \qquad \Rightarrow \qquad \frac{d\mathbf{t}}{ds} = \kappa \mathbf{n}$$

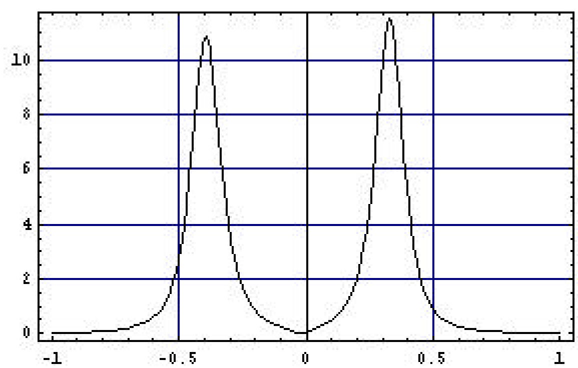
where κ is the curvature of the curve, and **n** is the unit normal vector to the curve. To compute κ, we use

$$\frac{d\mathbf{t}}{ds} = \frac{dx}{ds}\frac{d\mathbf{t}}{dx} = \frac{1}{\sqrt{1 + \left(\frac{df}{dx}\right)^2}} \frac{d}{dx}\left(\frac{1}{\sqrt{1 + \left(\frac{df}{dx}\right)^2}} \frac{d\mathbf{r}}{dx} \right) , \quad \kappa = \sqrt{\frac{d\mathbf{t}}{ds} \bullet \frac{d\mathbf{t}}{ds}} = \left\| \frac{d\mathbf{t}}{ds} \right\| .$$



6. Compute the curvature of the curve defined by the above data.

The curvature κ is defined in the previous question.



7. What is the Gauss-Legendre quadrature (numerical integration) ?

In order to integrate a function $f(s)$ defined on an interval (-1, +1), we apply the quadrature using the n number of quadrature points s_i , $i = 1,\ldots,n$, and the n number of weights w_i , $i = 1,\ldots,n$:

$$\int_{-1}^{1} f(s) ds \approx \sum_{i=1}^{n} w_i f(s_i)$$

where the quadrature points s_i are the roots of the n degree Legendre polynomial $L_n(s_i) = 0$, $s_i \in (-1, 1)$ that is a n degree polynomial. The weights w_i are obtained so as to

$$\int_{-1}^{1} x^I \, dx = \sum_{i=1}^{n} w_i f(s_i) \quad , \quad I = 0,1,\ldots.,n-1$$

Using this quadrature, we can integrate exactly up to $2n-1$ degree polynomials.

8. Obtain the first variation of the following functionals at u in the direction v :

(1) $$F(v) = \frac{1}{2}\int_0^L \left\{ EA\left(\frac{dv}{dx}\right)^2 + kv^2 - fv \right\} dx + \frac{1}{2}\sum_{i=1}^{m} \left\{ k_i v(x_i)^2 - f_i v(x_i) \right\}$$

$$\begin{aligned}
\delta F(u)(v) &= \lim_{\alpha \to 0} \frac{\partial F(u+\alpha v)}{\partial \alpha} \\
&= \lim_{\alpha \to 0} \frac{1}{2}\int_0^L \frac{\partial}{\partial \alpha} \left\{ EA\left(\frac{du}{dx} + \alpha \frac{dv}{dx}\right)^2 + k(u+\alpha v)^2 - f(u+\alpha v) \right\} dx \\
&+ \lim_{\alpha \to 0} \frac{1}{2}\sum_{i=1}^{m} \left\{ k_i \{(u+\alpha v)(x_i)\}^2 - f_i (u+\alpha v)(x_i) \right\} \\
&= \lim_{\alpha \to 0} \int_0^L \left\{ EA\left(\frac{du}{dx} + \alpha \frac{dv}{dx}\right)\frac{dv}{dx} + k(u+\alpha v)v - \frac{1}{2} f(v) \right\} dx \\
&+ \lim_{\alpha \to 0} \sum_{i=1}^{m} \left\{ k_i (u+\alpha v)(x_i) v(x_i) - \frac{1}{2} f_i v(x_i) \right\} \\
&= \int_0^L \left\{ EA \frac{du}{dx}\frac{dv}{dx} + kuv - \frac{1}{2} fv \right\} dx + \sum_{i=1}^{m} \left\{ k_i u(x_i) v(x_i) - \frac{1}{2} f_i v(x_i) \right\}
\end{aligned}$$

(2) $$F(v) = \frac{1}{2}\int_\Omega \left(\nabla v^T \mathbf{k} \nabla v + k_0 v^2 \right) d\Omega - \int_\Omega fv \, d\Omega$$

$$\delta F(u)(v) = \int_\Omega \left(\nabla v^T \frac{1}{2}\left(\mathbf{k} + \mathbf{k}^T\right)\nabla u + k_0 uv \right) d\Omega - \int_\Omega fv \, d\Omega$$

(3) $F(\mathbf{v}) = \frac{1}{2}\int_\Omega \boldsymbol{\varepsilon}(\mathbf{v})^T \mathbf{D}\, \boldsymbol{\varepsilon}(\mathbf{v}) d\Omega - \int_\Omega \mathbf{v}^T \mathrm{r}\, \mathbf{b} d\Omega$, where

$$\boldsymbol{\varepsilon}(\mathbf{v}) = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{bmatrix} \begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix} \quad , \quad \mathbf{v} = \begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix} \quad , \quad \mathbf{b} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \end{Bmatrix}$$

and $\mathbf{D}$ is 6-by-6 symmetric matrix.

$$\mathrm{d}F(\mathbf{u})(\mathbf{v}) = \int_\Omega \boldsymbol{\varepsilon}(\mathbf{v})^T \mathbf{D}\, \boldsymbol{\varepsilon}(\mathbf{u}) d\Omega - \int_\Omega \mathbf{v}^T \mathrm{r}\, \mathbf{b} d\Omega$$

9. Find the necessary condition of the constrained minimization problem

$$\min_{v \in K} \quad F(v) \quad , \quad K = \{ v \in V | \quad v - g \le 0 \quad in \quad (0, L) \ \}$$

for the functional defined in Problem 8-(1).

Suppose that u is a minimizer of the functional F on K. Then we have

$$u \in K \quad : \quad F(w) \ge F(u) \quad , \quad \forall w \in K$$

Noting that the constrained set K is convex. Taking

$$w = (1 - \mathrm{a})u + \mathrm{a} v = u + \mathrm{a}(v - u) \in K \ , \ \forall v \in K ,$$

we have

$$F\left(u+\alpha(v-u)\right)-F(u)\geq 0 \quad , \quad \forall v\in K$$
$$\lim_{\alpha\to 0}\frac{F\left(u+\alpha(v-u)\right)-F(u)}{\alpha}=\lim_{\alpha\to 0}\frac{\partial}{\partial\alpha}F\left(u+\alpha(v-u)\right)\geq 0 \quad , \quad \forall v\in K$$

that is

$$\delta F(u)(v-u)=\lim_{\alpha\to 0}\frac{\partial F\left(u+\alpha(v-u)\right)}{\partial\alpha}$$
$$=\int_0^L\left\{EA\frac{du}{dx}\frac{d}{dx}(v-u)+ku(v-u)-\frac{1}{2}f(v-u)\right\}dx$$
$$+\sum_{i=1}^{m}\left\{k_i u(x_i)(v-u)(x_i)-\frac{1}{2}f_i(v-u)(x_i)\right\}\geq 0 \quad , \quad \forall v\in K$$

10. Define the Haar Wavelet.

Based on the "mother wavelet" function

$$\psi(x)=\begin{cases}+1 & , \quad 0\leq x<\frac{1}{2}\\ -1 & , \quad \frac{1}{2}<x\leq 1\\ 0 & \quad otherwise\end{cases}$$

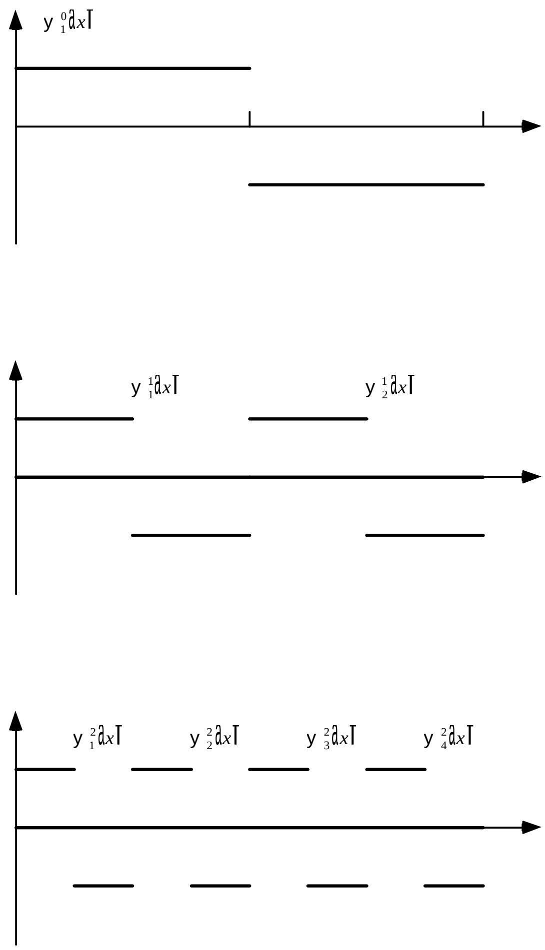
we form the mutually orthogonal wavelet functions:

$$\psi_k^j(x)=\psi\left(2^j x-k+1\right) \quad , \quad k=1,.....,2^j \quad , \quad j=0,1,2,......$$

It is clear that

$$\left(\psi_k^j,\psi_{\bar{k}}^{\bar{j}}\right)=\int_{-1}^{1}\psi_k^j(x)\psi_{\bar{k}}^{\bar{j}}(x)dx=0 \quad , \quad \forall k\neq\bar{k} \quad , \quad \forall j\neq\bar{j}$$

that is, they are orthogonal. Furthermore, $\psi_k^{j+1}(x)$ has twice more resolution, that is a half interval of $\psi_k^j(x)$.



11. Define the multi-resolution analysis.

As basis functions for approximation of a boundary value problem, we shall apply the special form defined by

$$\phi_{2^{j-1}+k}(x) = \psi_k^j(x) = \psi\left(2^j x - k + 1\right) \quad , \quad k = 1, \ldots\ldots, 2^j \;\; , \;\; j = 0,1,2,\ldots\ldots$$

so that the approximation in term of the index j+1 involves the twice more terms than that of the case j, that is, twice more resolution is introduced in the approximation by an increase of the range of the index j. Thus, by increasing the range of the index j, we have multiple resolution in the approximation, and it is called the multi-resolution analysis. This may be represented by the following form:

$$u(x) \approx u_J(x) = \sum_{j=0}^{J} \sum_{k=1}^{2^j} c_{2^{j-1}+k} \mathrm{f}_{2^{j-1}+k}(x) \supset u_{\bar{J}}(x) \quad , \quad J > \bar{J} \, .$$

The approximation $u_{J+1}(x)$ contains 2^{J+1} terms more basis functions than $u_J(x)$, and it has twice more resolution than $u_J(x)$.

You should also review the problems which were suggested in 1997 Fall term.