GLOBAL-LOCAL APPROXIMATIONS

Puneet Singla

MAE 502

Department of Mechanical & Aerospace Engineering
University at Buffalo

http://www.acsu.buffalo.edu/psingla/MAE502/

November 12, 2007
What Are We Looking For?

- Interpolation.
- Local support
- Interelement continuity/compatibility
- Completeness.
- Choices: FEM, MLS, Splines, RBF etc.

Most Important Characteristic: *Independent Local Approximations*
What Are We Looking For?

\[ f_1(t) = f_{10}e^{-\alpha t} \]

\[ f_2(t) = f_{20} + \beta t \]

\[ f_3(t) = f_{30} + \gamma \cos(\omega t - \phi) \]
WHAT ARE WE LOOKING FOR?

Local Approximations

Resulting Curve

A
B
C

Time $t$

$f(t)$
AVERAGING PROCESS

\[
\bar{F}_{i-1}(X) = w(i-1)F_{i-1}(X) + w(i)F_i(X)
\]

\[
\bar{F}_i(X) = w(i)F_i(X) + w(i+1)F_{i+1}(X)
\]

\[
\bar{F}_{i+1}(X) = w(i+1)F_{i+1}(X)
\]

\[
x^i = (X - x^i) / h; \quad |x^i| \leq 1
\]
The first derivative of the weighting function must have an $m^{th}$-order osculation with $w(0) = 1$ at the centroid of its respective local approximation.

$$w(0) = 1 \quad d^k w \bigg|_{x=0} = 0 \quad k = 0, 1, \ldots, m \quad (3.4)$$

The weighting function must have an $(m+1)^{th}$-order zero at the centroid of its neighboring local approximation.

$$w(1) = 0 \quad d^k w \bigg|_{x=1} = 0 \quad k = 0, 1, \ldots, m \quad (3.5)$$

The sum of two neighboring weighting functions must be unity over the entire closed interval.

$$w(Ix) + w(Ix - 1) = 1 \quad \forall x, -1 \leq x \leq 1 \quad (3.6)$$
The first derivative of the weighting function must have an $m^{th}$-order osculation with $w(0) = 1$ at the centroid of its respective local approximation.

$$w(0) = 1$$

$$\frac{d^k w}{dx^k}|_{x=0} = 0 \quad k = 0, 1, \cdots, m \quad (3.4)$$

The weighting function must have an $(m + 1)^{th}$-order zero at the centroid of its neighboring local approximation.

$$w(1) = 0$$

$$\frac{d^k w}{dx^k}|_{x=1} = 0 \quad k = 0, 1, \cdots, m \quad (3.5)$$

The sum of two neighboring weighting functions must be unity over the entire closed interval.

$$w(lx) + w(lx - 1) = 1 \quad \forall x, -1 \leq x \leq 1 \quad (3.6)$$
The first derivative of the weighting function must have an $m^{th}$-order osculation with $w(0) = 1$ at the centroid of its respective local approximation.

$$w(0) = 1$$
$$\frac{d^k w}{dx^k}|_{x=0} = 0 \quad k = 0, 1, \cdots, m \quad (3.4)$$

The weighting function must have an $(m+1)^{th}$-order zero at the centroid of its neighboring local approximation.

$$w(1) = 0$$
$$\frac{d^k w}{dx^k}|_{x=1} = 0 \quad k = 0, 1, \cdots, m \quad (3.5)$$

The sum of two neighboring weighting functions must be unity over the entire closed interval.

$$w(lx) + w(lx - 1) = 1 \quad \forall x, \ -1 \leq x \leq 1 \quad (3.6)$$
We assume the following particular form for weighting function,

\[ w(x) = 1 - J(x) \]  

(1)

where, \( J(x) \) is a polynomial in the independent variable whose first derivative is given by following expression:

\[ \frac{dJ(x)}{dx} = Cx^m(1-x)^m \]  

(2)

**NOTE:** This particular form for the weighting function is in accordance with the fact that *first m partial derivatives of the weighting function vanishes at end points and \( w(0) = 1 \).*

The remaining boundary conditions on the weighting function, \( w(x) \), will completely define the constants in Eq. (2).
Boundary Value Solution for Weighting Functions

- We have:

\[ J(1) = C \int_{0}^{1} x^m (1 - x)^m \, dx = 1 \]  \hspace{1cm} (3)

That means the appropriate value for constant, \( C \), is given by:

\[ C = \left[ \int_{0}^{1} x^m (1 - x)^m \, dx \right]^{-1} = \frac{(2m + 1)!}{(m!)^2} \]  \hspace{1cm} (4)

- Thus, the general form for weighting function can now be written as:

\[ w(x) = 1 - \frac{(2m + 1)!}{(m!)^2} \int_{0}^{x} x^m (1 - x)^m \, dx \]  \hspace{1cm} (5)
Further, the binomial theorem allows us to expand the above integrand:

\[ w(x) = 1 - \frac{(2m+1)!}{(m!)^2} \int_0^x \sum_{n=0}^m \binom{m}{n} x^n x^{m-n} (-1)^n \, dx \]  

(6)

Now, integrating the above expression term by term yield the following expression for weighting function, \( w(x) \):  

\[ w(x) = 1 - K \sum_{n=0}^m A_n x^{2m-n+1} \]  

(7)

where, \( K \) and \( A_n \) are given by following expressions:

\[ K = \frac{(2m+1)!(-1)^m}{(m!)^2} \quad A_n = \frac{(-1)^n m! C_n}{2m-n+1} \]  

(8)

Finally, to obtain the expression for weighting function in interval \([-1, 1]\) instead of \([0, 1]\) the absolute value of \( x \) is used as independent variable than \( x \).
**Weighting Functions**

**Table:** Weight functions for higher order continuity.

<table>
<thead>
<tr>
<th>order of piecewise continuity</th>
<th>Weight Function:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0$</td>
<td>$w(x) = 1 - y$</td>
</tr>
<tr>
<td>$1$</td>
<td>$w(x) = 1 - y^2(3 - 2y)$</td>
</tr>
<tr>
<td>$2$</td>
<td>$w(x) = 1 - y^3(10 - 15y + 6y^2)$</td>
</tr>
<tr>
<td>$3$</td>
<td>$w(x) = 1 - y^4(35 - 84y + 70y^2 - 20y^3)$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$m$</td>
<td>$w(x) = 1 - y^{m+1} \left{ \frac{(2m+1)!(-1)^m}{(m!)^2} \sum_{k=0}^{m} \frac{(-1)^k}{2m-k+1} \binom{m}{k} y^{m-k} \right}$</td>
</tr>
</tbody>
</table>
Averaging Process

Preliminary Approximations:

\[ F_{11}(x, y) \]

\[ F_{01}(x, y) \]

\[ F_{00}(x, y) \]

\[ F_{10}(x, y) \]

Weight functions:

\[ w_{11}(x, y) = x^2(3 - 2x)y^2(3 - 2y) \]

\[ w_{01}(x, y) = w_{11}(1 - x, y) \]

\[ w_{00}(x, y) = w_{11}(1 - x, 1 - y) \]

\[ w_{10}(x, y) = w_{11}(x, 1 - y) \]

Final Approximation:

\[ \bar{F}(x, y) = \sum_{i=0}^{1} \sum_{j=0}^{1} w_{ij}(x, y) F_{ij}(x, y) \]

valid over \( \{0 \leq x \leq 1, 0 \leq y \leq 1\} \)

Weight functions are a partition of unity:

\[ \sum_{i=0}^{1} \sum_{j=0}^{1} w_{ij}(x, y) = 1 \]
WEIGHTING FUNCTIONS (FOR APPROXIMATION IN 2-D)
The preliminary approximations, while arbitrary, in particular could be chosen to minimize the least square criterion:

\[ J = \frac{1}{2} \int_{-1}^{1} w(x) [F(X) - F_I(X)]^2 dx \]  \hspace{1cm} (3.21)

If \( F_I(X) \) is considered as a linear combination of a set of linearly independent basis functions:

\[ F_I(X) = \sum_{i=0}^{n} a_i \phi_i(x); \quad X = I X + h x \]  \hspace{1cm} (3.22)

The least square criterion gives the optimum (minimum integral least square fit error) coefficients as: \( a = M^{-1} c \)
The preliminary approximations, while arbitrary, in particular could be chosen to minimize the least square criterion:

\[ J = \frac{1}{2} \int_{-1}^{1} w(x)[F(X) - F_I(X)]^2 dx \] \hspace{1cm} (3.21)

If \( F_I(X) \) is considered as a linear combination of a set of linearly independent basis functions:

\[ F_I(X) = \sum_{i=0}^{n} a_i \phi_i(x); \ X = Ix + hx \] \hspace{1cm} (3.22)

The least square criterion gives the optimum (minimum integral least square fit error) coefficients as: \( a = M^{-1}c \)
The preliminary approximations, while arbitrary, in particular could be chosen to minimize the least square criterion:

\[
J = \frac{1}{2} \int_{-1}^{1} w(x) [F(X) - F_l(X)]^2 dx \quad (3.21)
\]

If \( F_l(X) \) is considered as a linear combination of a set of linearly independent basis functions:

\[
F_l(X) = \sum_{i=0}^{n} a_i \phi_i(x); \quad X = Ix + hx \quad (3.22)
\]

The least square criterion gives the optimum (minimum integral least square fit error) coefficients as: \( a = M^{-1} c \) with,

\[
c_j = < F(x), \phi_j(x) >
\]

\[
M_{ij} = < \phi_i, \phi_j >
\]
The preliminary approximations, while arbitrary, in particular could be chosen to minimize the least square criterion:

$$J = \frac{1}{2} \int_{-1}^{1} w(x)[F(X) - F_{I}(X)]^2 dx \quad (3.21)$$

If $F_{I}(X)$ is considered as a linear combination of a set of linearly independent basis functions:

$$F_{I}(X) = \sum_{i=0}^{n} a_{i}\phi_{i}(x); \quad X = IX + hx \quad (3.22)$$

The least square criterion gives the optimum (minimum integral least square fit error) coefficients as: $a = M^{-1} c$

If basis functions satisfy the orthogonality condition:

$$< \phi_{i}(x), \phi_{j}(x) > = k_{i}\delta_{ij}, \quad k_{i} \Delta = \mu_{ii}$$

Then least square solution simplifies to the simple uncoupled result to compute the Fourier coefficients:

$$a_{i} = \frac{< F(x), \phi_{i}(x) >}{k_{i}}, \quad i = 1, 2, \cdots, n \quad (3.30)$$
Preliminary Approximations
(The generalization to $N$-dimensions is really, really beautiful!)

(a) 1-D Basis Functions

$$w(.) = \prod_{i=0}^{N} \left[ 1 - x_i^2 (3 - 2|x_i|) \right], \quad \phi_{i_1 \ldots i_N}(.) = \prod_{i=0}^{N} \phi_i(x_i)$$

(b) 2-D Basis Functions

$$\alpha_{i_1 \ldots i_N} = \frac{\langle \phi_{i_1 \ldots i_N}(x_1, \ldots, x_N), F(X_1, \ldots, X_N) \rangle}{\prod_{j=1}^{N} k_j} \quad (3.45)$$
Basic Idea
Basic Idea
Basic Idea
The main steps of the GLO-MAP algorithm are as follows:

- Choose a set of sequential neighboring points, $I_X$, arbitrary in number and location. These points serve as the centroids of validity for the local functional approximation, $F_l$.
  - The density and location of these points depends upon numerous factors like location and density of available measurement data and desired degree of approximation.
The main steps of the GLO-MAP algorithm are as follows:

- Choose set of basis functions based on computational efficiency or a priori knowledge of the nature of the given input-output data to approximate $F(X)$ in local neighborhood of a centroid of validity, $lX$.
  - One attractive choice for the basis functions is the orthogonal polynomial basis functions as discussed earlier.
The main steps of the GLO-MAP algorithm are as follows:

- Determine the Fourier coefficients corresponding to each local approximation.
  - For the approximation of a given continuous functional form or from dense discretely measurable functions, the Fourier coefficients can be computed by \textit{numerically evaluating the integral expressions} using standard numerical integration algorithms.
The main steps of the GLO-MAP algorithm are as follows:

In case of the GLO-MAP algorithm the numerical integration procedure can be summarized as below:

1. Determine Gaussian quadrature points, $X_G$, in unit hypercube.

2. For each quadrature point, determine the measurement points $X_i$, for which weight function has non-zero value.

3. Determine function value at quadrature point, $F(X_G)$ can be taken as a weighted average of function value at $X_i$.

$$F(X_G) = \frac{1}{\sum_i w(X_i)} \sum_i F(X_i) w(X_i)$$ (9)

4. Evaluate the numerical integrals.
The main steps of the GLO-MAP algorithm are as follows:

- It should be noted that Gaussian quadrature points are the same for each local approximation and can be pre-computed.
  - However, the numerical evaluation of integral expression of several variables, over regions with dimension greater than one, is not easy.
  - As a rule of thumb, the number of function evaluations needed to sample an $N$-dimensional space increases as the $N^{th}$ power of the number needed to do a one-dimensional integral, resulting in increased computational cost associated with numerical integration proportion to the $N^{th}$ power.
- Use discrete orthogonal basis functions.
The main steps of the GLO-MAP algorithm are as follows:

- The final step of the GLO-MAP algorithm is the use of weighting functions to merge the local approximations into a single $m^{th}$ order continuous functional model.

- Note, this step is the most important feature of the GLO-MAP algorithm as it reduces the systematic errors introduced due to the neglected interaction between different local models by blending overlapping local approximations into a global one.
Algorithm Implementation

Input-Output Data

Choose a set of sequential points as centroids for local approximations

Choose set of orthogonal basis functions

For each centroid point, find the measurement data points which lies in the local neighborhood of the centroid

Determine Fourier coefficients pertaining to each local approximation

Merge different local approximations using weighting functions given in Table 1

Compute measure of approximation error by computing the global covariance matrix by merging covariance matrix corresponding to different local approximations.

Refine the distance between different centroid points.

Is approximation error less than pre-specified tolerance?

Include higher order basis functions

Stop

Figure: Flowchart for the GLO-MAP based multi-resolution algorithm.
There are many engineering application problems which need to be solved in an iterative manner in real-time by successively approximating the input-output data.

- Kalman filter algorithm can be used to update a priori learned GLO-MAP network parameters in real time whenever new measurements are available.

- The main challenge associated with the use of the Kalman filter in the GLO-MAP algorithm is the dynamic state vector i.e. the components of state vector of Kalman filter changes with every measurement data depending upon the location of measurement data relative to centroids of validity of local approximations.

- However, when a new measurement is available, only the neighboring $2^N$ local approximations (associated with the $2^N$ vertices of the hypercube containing the new measurement) will have non-zero contribution in final global map.
Sequential Version

The main steps involved in the implementation of sequential version of the GLO-MAP algorithm are as follows:

- Choose a set of sequential neighboring points, \( I_X \), arbitrary in number and location. These points serve as the centroids of validity for the local functional approximation, \( F_I \).
  - The density and location of these points depends upon numerous factors like location and density of available measurement data and desired degree of approximation.
The main steps involved in the implementation of sequential version of the GLO-MAP algorithm are as follows:

- **Choose a set of basis functions** (preferably orthogonal functions) to approximate $F(X)$ in the local neighborhood of a centroid of validity, $I_X$.
- Initialize the coefficients of each basis functions to be zero and associated covariance matrix ($P_G$) to be identity times a large number.
The main steps involved in the implementation of sequential version of the GLO-MAP algorithm are as follows:

- Given a new measurement data point, \((X, F(X))\), find the neighboring \(2^N\) centroids such that weight function associated with these \(2^N\) centroids have non-zero value at the measurement point.

- Include the coefficients of the basis functions associated with these centroids in the local algebraic Kalman filter state vector, denoted by \(x\). Let \(x_G\) denotes the global super-set of coefficients and \(M\) be a selection matrix consisting of zeros and ones that satisfies \(x = Mx_G\).

- Extract rows and columns of \(P_G\) corresponding to the coefficients associated with these \(2^N\) centroids and denote them by \(P_k^- = MP_GM^T\).
The main steps involved in the implementation of sequential version of the GLO-MAP algorithm are as follows:

- Use the following equations to compute the local Kalman gain and update state vector, $x$, and the associated covariance matrix, $P$.

\[
\begin{align*}
x_k^+ &= x_k^- + K(F(X) - \Phi(.x_k^-)) \\
P_k^+ &= (I - K\Phi(.))P_k^- \\
K &= P_k^- \Phi(.)^T(\Phi(X)P_k^- \Phi(.)^T + R_k)^{-1}
\end{align*}
\]

Matrices $\Phi$ and $R$ are given by following equations:

\[
\begin{align*}
\Phi_k &= \begin{bmatrix} \phi_1(X_k) & \cdots & \phi_N(X_k) \end{bmatrix} \\
R_k &= \sigma_w(X_k)
\end{align*}
\]
The main steps involved in the implementation of sequential version of the GLO-MAP algorithm are as follows:

- Update rows and columns of the *global covariance matrix, $P_G$* and coefficients associated with each local approximation.

- Once the value of $P_G$ is less than a pre-specified tolerance, use appropriate weighting functions to merge various local approximations into a single $m^{th}$ order continuous functional model.
RESULTS

In first test case, the continuous functional expression is used to obtain the coefficients of the preliminary local approximations.

In second test case a discrete measurement data is used to compute preliminary approximations.

- 100 random samples over the interval [0-10, 0-10] for both $X_1$ and $X_2$, giving total $10^4$ measurements.
RESULTS

(c) True Surface

(d) True Contours

(e) Orthogonalization Error

(f) Approximation Error
RESULTS

(g) True Surface

(h) True Contours

(i) Error Surface

(j) Error Contours
(k) True porkchop plot for departure $\Delta V_\infty$

(l) True porkchop plot for arrival $\Delta V_\infty$

(m) True surface plot for departure $\Delta V_\infty$

(n) True surface plot for arrival $\Delta V_\infty$
PORK-CHOP PLOT APPROXIMATION

(a) Approximated porkchop plot for departure $\Delta V_{\infty}$
(b) Approximated porkchop plot for arrival $\Delta V_{\infty}$

(c) Approximated surface plot for departure $\Delta V_{\infty}$
(d) Approximated surface plot for arrival $\Delta V_{\infty}$
PORK-CHOP PLOT APPROXIMATION

(a) Error contour plot for $220 \leq \text{TOF} \leq 300$

(b) Error contour plot for $\text{TOF} < 220$

(c) Error contour plot for $\text{TOF} > 300$

(d) Error contour plot for whole domain
(e) Error contour plot for $220 \leq TOF \leq 300$

(f) Error contour plot for $TOF < 220$

(g) Error contour plot for $TOF > 300$

(h) Error contour plot for whole domain
Pork-Chop Plot Approximation

- Original measurement points for approximation purposes: 
  \[
  3 \times (365 \times 246) = 269370.
  \]

- GLO-MAP approximation: 
  \[
  8 \times (625 + 2.400) = 11400 \text{ real numbers}.
  \]