MPI Summer School 2024
Lecture 1: Intro to DG Methods

Francis X. Giraldo

Department of Applied Mathematics
Naval Postgraduate School
Monterey CA. 93943-5216

September 2-6, 2024

Table of contents

Outline

- Overview of existing methods
- Discontinuous Galerkin method
- 1D Interpolation
- 1D Integration

Overview of Methods

- Differential Form
  1. Finite Difference Methods
  2. Spectral Multi-domain Penalty Methods
- Integral Form
  1. Finite Element Methods
  2. Finite Volume Methods
  3. Spectral Element Methods
  4. Discontinuous Galerkin Methods
  5. Spectral Methods

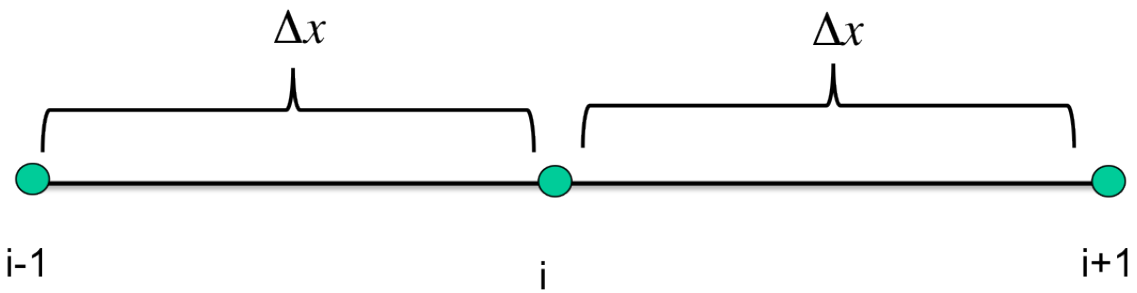
Differential Form

- Let's consider the canonical equation:

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} = 0$$

  where $q = q(x, t)$, $f = f(x, t)$, and $f = qu$ where we can assume that u is constant for now.

- on the simple 1D grid:

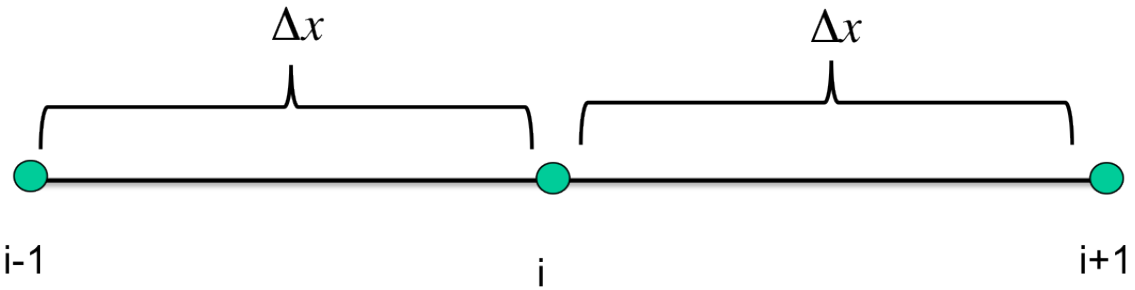


Taylor Series Expansion

- Beginning with the PDE:

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} = 0$$

- we can use Taylor series expanding about the point i (x_i)



- to generate the following approximations

$$f_{i+1} = f_i + \Delta x \frac{\partial f_i}{\partial x} + \frac{\Delta x^2}{2} \frac{\partial^2 f_i}{\partial x^2} + \mathcal{O}(\Delta x^3)$$

$$f_{i-1} = f_i - \Delta x \frac{\partial f_i}{\partial x} + \frac{\Delta x^2}{2} \frac{\partial^2 f_i}{\partial x^2} - \mathcal{O}(\Delta x^3)$$

Taylor Series Expansion

- subtracting these two relations gives

$$f_{i+1} - f_{i-1} = 2\Delta x \frac{\partial f_i}{\partial x} + \mathcal{O}(\Delta x^3)$$

- which then allows us to write:

$$\frac{f_{i+1} - f_{i-1}}{2\Delta x} = \frac{\partial f_i}{\partial x} + \mathcal{O}(\Delta x^2).$$

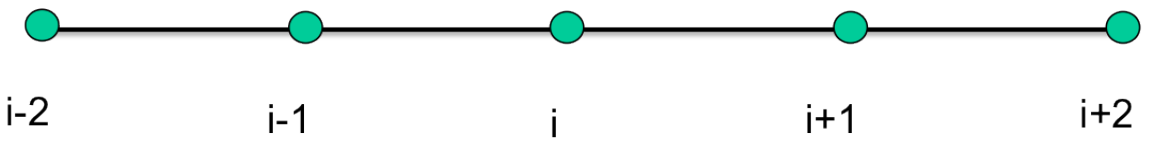
Taylor Series Expansion

- If a higher order approximation is desired we then require information from additional grid points. For example:

$$f_{i+1} = f_i + \Delta x \frac{\partial f_i}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 f_i}{\partial x^2} + \frac{\Delta x^3}{6}\frac{\partial^3 f_i}{\partial x^3} + \mathcal{O}(\Delta x^4),$$

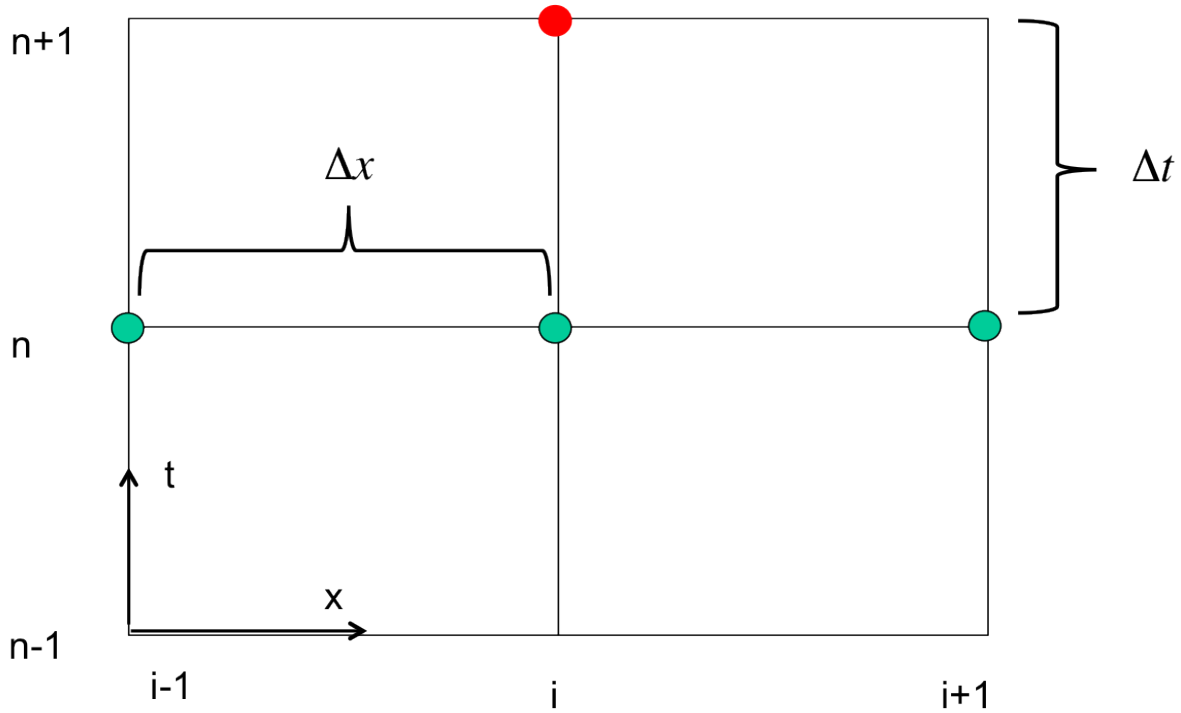
$$f_{i-1} = f_i - \Delta x \frac{\partial f_i}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 f_i}{\partial x^2} - \frac{\Delta x^3}{6}\frac{\partial^3 f_i}{\partial x^3} + \mathcal{O}(\Delta x^4),$$

$$f_{i+2} = f_i + 2\Delta x \frac{\partial f_i}{\partial x} + \frac{(2\Delta x)^2}{2}\frac{\partial^2 f_i}{\partial x^2} + \frac{(2\Delta x)^3}{6}\frac{\partial^3 f_i}{\partial x^3} + \mathcal{O}(\Delta x^4)$$



X-T Diagram

- Lets now introduce the following x-t diagram:



Taylor Series in X and T

- Let's look at the following expansions about the point (x_i, t^n)

$$\begin{aligned} q_i^n &\equiv q(x_i, t^n) \\ q_i^{n+1} &\equiv q(x_i, t^n + \Delta t) = q(x_i, t^n) + \Delta t \frac{\partial q(x_i, t^n)}{\partial t} + \mathcal{O}(\Delta t^2) \end{aligned}$$

- and these:

$$\begin{aligned} f_i^n &\equiv f(x_i, t^n) \\ f_{i-1}^n &\equiv f(x_i - \Delta x, t^n) = f(x_i, t^n) - \Delta x \frac{\partial f(x_i, t^n)}{\partial x} + \mathcal{O}(\Delta x^2) \end{aligned}$$

Difference Stencil

- when substituted into the 1D wave equation yields (where we have dropped subscripts and superscripts)

$$\frac{q(x,t) + \Delta t \frac{\partial q(x,t)}{\partial t} + \mathcal{O}(\Delta t^2) - q(x,t)}{\Delta t}$$
$$+ \frac{f(x,t) - f(x,t) + \Delta x \frac{\partial f(x,t)}{\partial x} + \mathcal{O}(\Delta x^2)}{\Delta x} = 0.$$

- Simplifying this expression yields, as $(\Delta x, \Delta t) \to 0$

$$\frac{\partial q(x,t)}{\partial t} + \frac{\partial f(x,t)}{\partial x} + \mathcal{O}(\Delta t, \Delta x) = 0$$

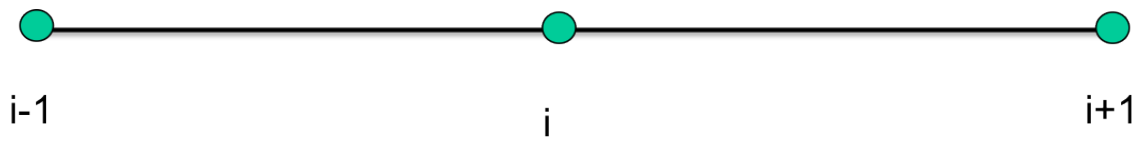
- In general one can write an approximation to the PDE as follows:

$$\frac{q_i^{n+1} - q_i^n}{\Delta t} + \sum_{k=-N}^{N} \frac{\beta_k f_{i+k}^n}{\Delta x} = 0$$

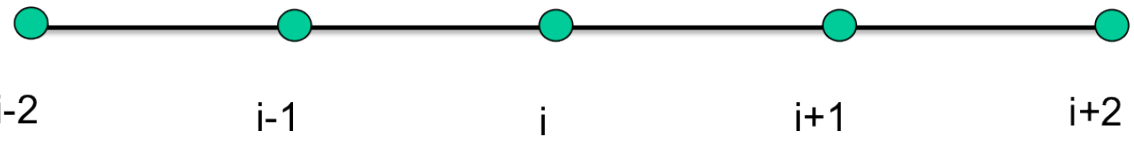
where N controls the size of the stencil and is $\mathcal{O}(2N)$.

Difference Stencil

- E.g., this 3 point stencil will produce a 2nd order approximation



- This 5 point stencil will produce a 4th order approximation



Matrix-Vector form of Difference Stencil

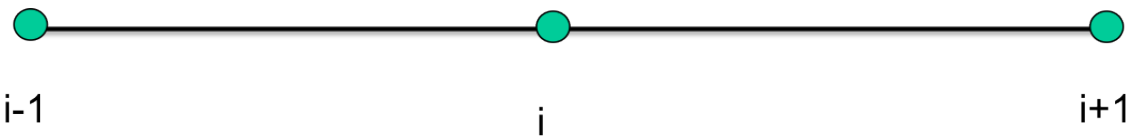
- For Euler in time and upwind in space, the stencil is:

$$\frac{q_i^{n+1} - q_i^n}{\Delta t} + \frac{f_i^n - f_{i-1}^n}{\Delta x} = 0$$

- Note that the approximation of the spatial derivative can be written as a matrix as follows:

$$\frac{q_i^{n+1} - q_i^n}{\Delta t} + D_{i,j} f_j^n = 0$$

- Next, assuming a grid comprised of 3 grid points



- We get the following matrix and vector (periodicity)

$$D_{i,j} = \begin{pmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}, \quad f_j^n = \begin{pmatrix} f_{i-1}^n \\ f_i^n \\ f_{i+1}^n \end{pmatrix}$$

Integral Form

- Let us now turn to Integral Forms
- For the canonical equation

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} = 0$$

where $q = q(x,t)$, $f = f(x,t)$, and $f = qu$

- We begin by approximating the solution variables as follows

$$q_N(x,t) = \sum_{i=0}^{N} \psi_i(x) q_i(t)$$

with $f_N = f(q_N(x,t))$, q being the expansion coefficients, ψ the basis (polynomial) functions, and N the order of the polynomial.

Integral Form

- We can construct the following approximations

$$q_N(x,t) = \sum_{i=0}^{N} \psi_i(x) q_i(t) \quad \text{and} \quad \frac{\partial q_N}{\partial t}(x,t) = \sum_{i=0}^{N} \psi_i(x) \frac{dq_i}{dt}(t)$$

$$f_N(x,t) = \sum_{i=0}^{N} \psi_i(x) f_i(t) \quad \text{and} \quad \frac{\partial f_N}{\partial x}(x,t) = \sum_{i=0}^{N} \frac{d\psi_i}{dx}(x) f_i(t)$$

- Subbing into our PDE yields

$$\frac{\partial q_N}{\partial t} + \frac{\partial f_N}{\partial x} = r \neq 0$$

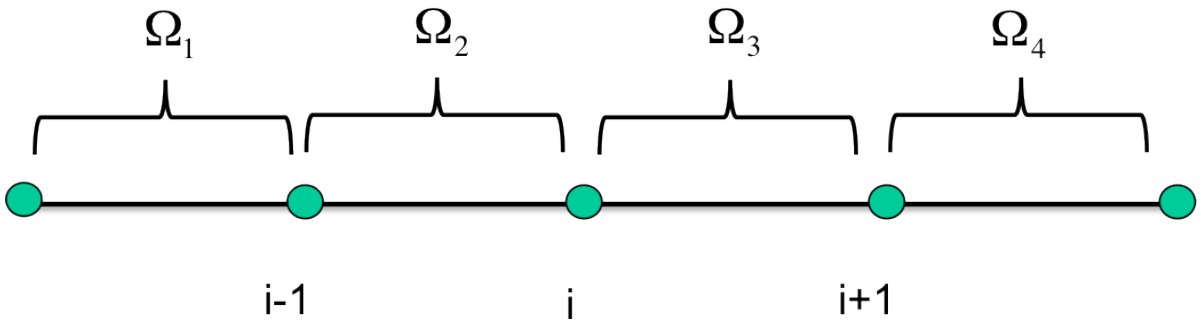
which is not equal to zero!.

Differential to Integral Form

- This is so because we have used a finite-dimensional approximation $q_N = \sum_{i=0}^{N} \psi_i q_i$.
- We resolve this issue by multiplying this approximation by a test function ψ and integrating to get

$$\int_{\Omega_e} \psi_i \frac{\partial q_N}{\partial t} d\Omega_e + \int_{\Omega_e} \psi_i \frac{\partial f_N}{\partial x} d\Omega_e = \int_{\Omega_e} \psi_i r \, d\Omega_e \equiv 0$$

- where the domain is partitioned as follows



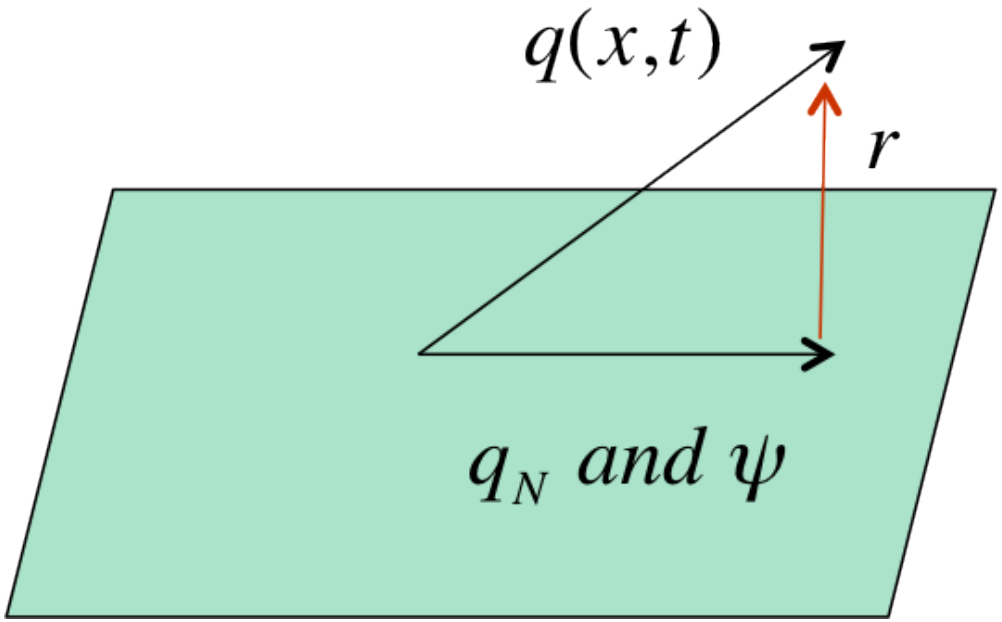
where $\Omega = \bigcup_{e=1}^{N_e} \Omega_e$ defines the total domain and $e = 1, ..., N_e$ are the elements.

Galerkin Projection

- The last integral in this expression

$$\int_{\Omega_e} \psi_i \frac{\partial q_N}{\partial t} \, d\Omega_e + \int_{\Omega_e} \psi_i \frac{\partial f_N}{\partial x} \, d\Omega_e = \int_{\Omega_e} \psi_i r \, d\Omega_e \equiv 0$$

  is zero because the residual r is orthogonal to ψ as follows:



- This is the case because ψ projects the solution q to this space (green plane).

Weak Integral Form

- Using calculus identities we can simplify the weak integral system into the form

$$\int_{\Omega_e} \psi_i \frac{\partial q_N}{\partial t} d\Omega_e + \int_{\Omega_e} \frac{\partial}{\partial x} (\psi_i f_N) \, d\Omega_e - \int_{\Omega_e} \frac{d\psi_i}{dx} f_N d\Omega_e = 0$$

- Integrating the second term (Fundamental Theorem of Calculus) gives

$$\int_{\Omega_e} \psi_i \frac{\partial q_N}{\partial t} d\Omega_e + [\psi_i f_N]_{\Gamma_e} - \int_{\Omega_e} \frac{d\psi_i}{dx} f_N d\Omega_e = 0$$

  where the term in the square brackets is evaluated at the boundary, Γ_e, of the domain Ω_e.

- Let's now describe the discontinuous Galerkin method.

Outline

- Overview of existing methods
- Discontinuous Galerkin method
- 1D Interpolation
- 1D Integration

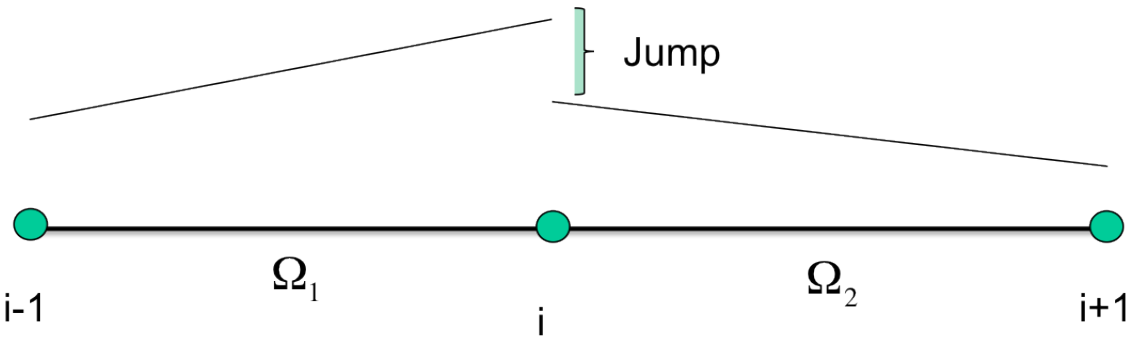
Discontinuous Galerkin Method

- The equation

$$\int_{\Omega_e} \psi_i \frac{\partial q_N}{\partial t} d\Omega_e + [\psi_i f_N]_{\Gamma_e} - \int_{\Omega_e} \frac{d\psi_i}{dx} f_N d\Omega_e = 0$$

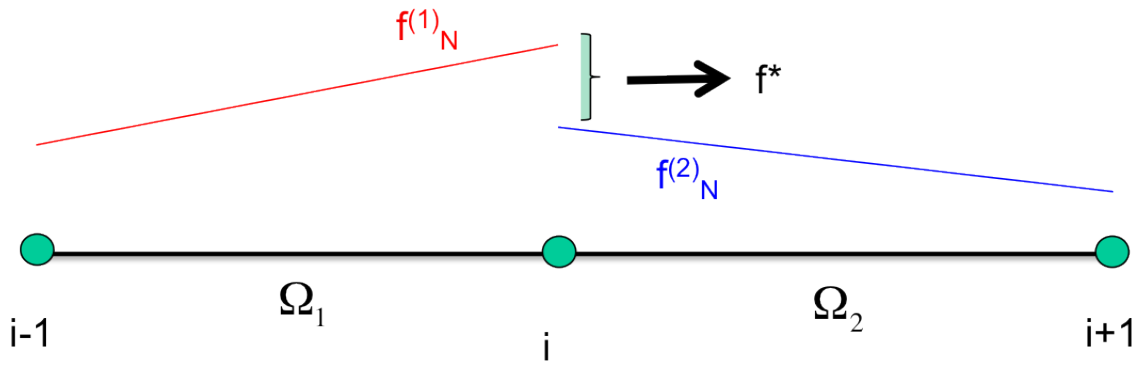
  represents the (weak) integral form of the original differential equation.

- The term $[\psi_i f_N]_{\Gamma_e}$ is what allows neighboring elements to communicate.



Numerical Flux

- However, depending on where we sample f_N we will get a different value (left element versus the right element).



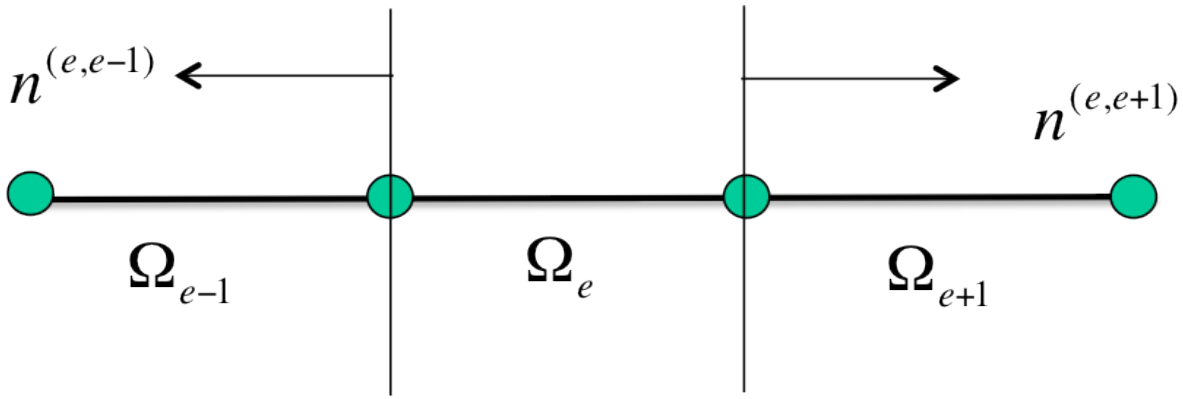
- To ensure conservation of all desired quantities (e.g., mass) we require the same flux that leaves one element to enter the neighbor.

Numerical Flux

- This is resolved by constructing a **numerical flux** that uses the values of both fluxes. I.e., we write

$$\int_{\Omega_e} \psi_i \frac{\partial q_N^{(e)}}{\partial t} d\Omega_e + \sum_{k=1}^{N_s} \widehat{\boldsymbol{n}}_{\Gamma_e}^{(e,k)} \psi_i f_N^{(*,k)} - \int_{\Omega_e} \frac{d\psi_i}{dx} f_N^{(e)} d\Omega_e = 0$$

where N_s denotes the number of side/edges ($N_s = 2$ in 1D) and $\widehat{\boldsymbol{n}}_{\Gamma_e}^{(e)}$ is the outward pointing normal vector (from element Ω_e).



Numerical Flux

- Examples of such numerical fluxes include standard **Riemann solvers**.
- A simple example is the Rusanov flux defined as

$$f_N^{(*,k)} = \frac{1}{2}\left[f_N^{(k)} + f_N^{(e)} - \lambda_{max}\widehat{\boldsymbol{n}}_{\Gamma_e}^{(e,k)}\left(q_N^{(k)} - q_N^{(e)}\right)\right]$$

  where $\lambda_{max} = |\frac{\partial f^{(e)}}{\partial q}, \frac{\partial f^{(k)}}{\partial q}|$ is the maximum wave propagation speed of the system.
- However, other fluxes used in FV methods can also be used (e.g., HLL, HLLC, Roe, etc.).

Finite Volume Limit

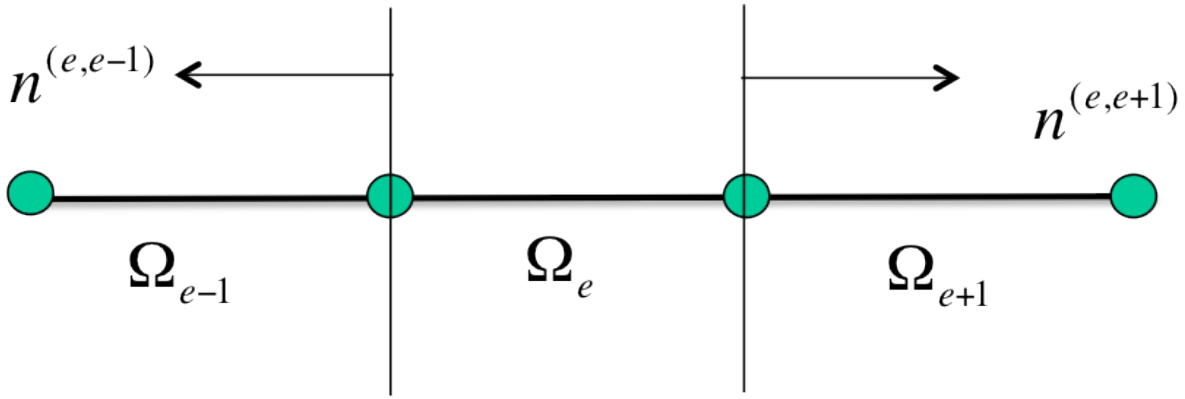
- Note that if we choose constant basis functions, say $\psi = 1$, we get

$$\int_{\Omega_e} \frac{\partial q_N^{(e)}}{\partial t} + \sum_{k=1}^{2} \widehat{\boldsymbol{n}}_{\Gamma_e}^{(e,k)} f_N^{(*,k)} = 0$$

- upon integrating, yields

$$\Delta x \frac{\partial q_N^{(e)}}{\partial t} + \left(\widehat{\boldsymbol{n}}_{\Gamma_e}^{(e,e+1)} f_N^{(*,e+1)} + \widehat{\boldsymbol{n}}_{\Gamma_e}^{(e,e-1)} f_N^{(*,e-1)} \right) = 0$$

- where the stencil is defined as follows



Finite Volume Limit

- where we have let $k = 1$ be the element $e - 1$ and $k = 2$ the element $e + 1$.
- Note that this gives the following normal vectors: $\widehat{\boldsymbol{n}}_{\Gamma_e}^{(e,e-1)} = -1$ and $\widehat{\boldsymbol{n}}_{\Gamma_e}^{(e,e+1)} = +1$
- Dividing by Δx gives the final form

$$\frac{\partial q_N^{(e)}}{\partial t} + \frac{f_N^{(*,e+1)} - f_N^{(*,e-1)}}{\Delta x} = 0$$

Finite Volume Limit

- Let us assume that $f = qu$ so that $\frac{\partial f}{\partial q} = u$ and so $\lambda_{max} = |u|$.
- If we further assume u is constant and moves from left to right then we can write $\lambda_{max} = u$ which we now use to rewrite the Rusanov flux as

$$f_N^{(*,e+1)} = \frac{1}{2}\left[f_N^{(e+1)} + f_N^{(e)} - u\left(q_N^{(e+1)} - q_N^{(e)}\right)\right] \equiv f^{(e)}$$
$$f_N^{(*,e-1)} = \frac{1}{2}\left[f_N^{(e-1)} + f_N^{(e)} + u\left(q_N^{(e-1)} - q_N^{(e)}\right)\right] \equiv f^{(e-1)}$$

- where we have used the fact that u is constant and so $f^{(k)} = q^{(k)}u$ for all elements k.

Finite Volume Limit

- Substituting the resulting flux into our difference stencil

$$\frac{\partial q_N^{(e)}}{\partial t} + \frac{f_N^{(*,e+1)} - f_N^{(*,e-1)}}{\Delta x} = 0$$

yields

$$\frac{\partial q_N^{(e)}}{\partial t} + \frac{f^{(e)} - f^{(e-1)}}{\Delta x} = 0$$

which is nothing more than a 1st order stencil typically associated with an upwinding finite volume method.

Outline

- Overview of existing methods
- Discontinuous Galerkin method
- 1D Interpolation
- 1D Integration

1D Interpolation

- So far we have seen that the difference between FV and DG methods is the use of the basis functions ψ.
- These basis functions are in fact interpolation functions. Let's define what we mean by this.
- Interpolation is the act of approximating a function $f(x)$ by an Nth degree interpolant I_N such that

$$I_N(f(x_i)) = f(x_i)$$

where x_i are $i = 0, ..., N$ specific points where the function is evaluated.

- Let us approximate the function $f(x)$ by the finite expansion

$$f(x) = \sum_{i=0}^{N} \phi_i(x)\widetilde{f}_i + e_N(x)$$

1D Interpolation

- where ϕ is the interpolating (approximating) function and is also called a basis function, $\widetilde{f}$ is the expansion coefficient, and e is the error incurred by only using N terms to approximate the function.
- Interpolation can be modal (spectral) or nodal (physical).
- Modal interpolation (called approximation) relies on orthogonal polynomials. This class of functions is usually associated with spectral methods.
- Nodal interpolation relies on Lagrange polynomials and the concept of cardinality. This class of functions is usually associated with finite element methods.

Modal Functions

- Let us call $\phi(x)$ the basis function that we wish to construct.
- One could construct Nth order polynomials by the following expansion

$$\phi_i(x) = x^i \; \forall \; i = 0, ..., N.$$

- The first three monomials are

$$\phi_0(x) = 1, \; \phi_1(x) = x, \; \phi_2(x) = x^2.$$

- This polynomial expansion has one very large deficiency - it is comprised of non-orthogonal polynomials.
- This is a big concern because it means that it does not form a complete basis (numerically speaking); that is, it is not guaranteed to be numerically linearly independent.
- If a set of polynomials are not linearly independent, then they can give you the same information at different points (redundancies or multiplicities).

Modal Functions

- While this issue will only arise for very large order it is best from the outset to do things in a mathematically rigorous way.
- Therefore, the best approach is to use orthogonal polynomials derived from the eigenvalue problem of the Sturm-Liouville operator on the domain of interest (e.g., $x \in [-1, +1]$).

Sturm-Liouville Operator

- In one space dimension, the Sturm-Liouville eigenvalue problem is written as

$$\frac{d}{dx}\left[\left(1-x^2\right)w(x)\frac{d}{dx}P_N^{(\alpha,\beta)}(x)\right]+\lambda_N w(x)P_N^{(\alpha,\beta)}(x)=0$$

  where $w(x)=(1-x)^\alpha(1+x)^\beta$ and $\lambda_N=N(N+\alpha+\beta+1)$ with $x\in[-1,+1]$.

- The importance of this equation is that its solution is the set of orthogonal polynomials with real eigenvalues; these orthogonal polynomials , $P_N^{(\alpha,\beta)}(x)$, are the Jacobi polynomials and are orthonormal with respect to the weighting function $w(x)$, that is, the following integral is satisfied

$$\int_{-1}^{+1} w(x)P_i^{(\alpha,\beta)}(x)P_j^{(\alpha,\beta)}(x)dx=\delta_{ij}\ \ \forall\,(i,j)=0,...,N.$$

Sturm-Liouville Operator

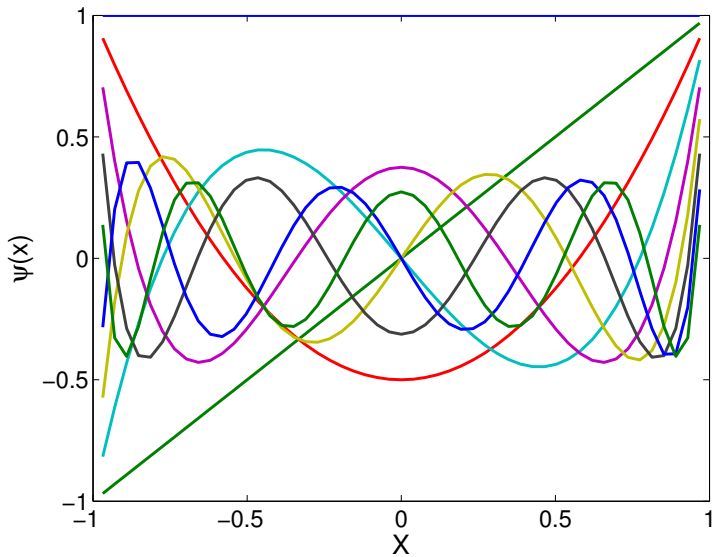
- These orthonormal functions form a complete basis in the function space defined by the boundary conditions, that is, the domain in which the Sturm-Liouville operator is solved.
- Note that the terms α and β in the Jacobi polynomials should be assumed to be integers for our purposes in constructing basis functions.

Remark

A good reason for using orthogonal polynomials is that they have simple recursion relations that allows us to construct arbitrarily high-order functions quite easily.

Legendre Polynomials

- Legendre polynomials are a special case of Jacobi polynomials and are defined as $L_N(x) = P_N^{(0,0)}(x)$.
- These polynomials are in fact the simplest Jacobi polynomials to consider and for this reason are typically the preferred choice for constructing orthogonal polynomials in the space $x \in [-1, +1]$.
- A set of 8th degree Legendre polynomials are given as follows



Legendre Polynomials: Example

- Let us now take a closer look at the first three Legendre polynomials

$$\phi_0(x) = 1,\ \phi_1(x) = x,\ \phi_2(x) = \frac{3}{2}x^2 - \frac{1}{2}.$$

- Note that if we used the monomial expansion and then applied Gram-Schmidt orthogonalization we would recover the Legendre polynomials (try it).
- If the function that we are trying to represent is $f(x) = a$ where a is a constant, then the approximation of this function is achieved by setting

$$\widetilde{f}_0 = a,\ \widetilde{f}_1 = 0, \text{ and } \widetilde{f}_2 = 0.$$

- Similarly, if $f(x) = bx$ then we would represent this function exactly with

$$\widetilde{f}_0 = 0,\ \widetilde{f}_1 = b, \text{ and } \widetilde{f}_2 = 0.$$

Legendre Polynomials: Summary

- For a general polynomial $f(x) = a + bx + cx^2$ we would represent this function exactly with

$$\widetilde{f}_0 = a + \frac{1}{3}c,\ \widetilde{f}_1 = b,\ \text{and}\ \widetilde{f}_2 = \frac{2}{3}c.$$

- From this very simple example, we can see that if $f(x)$ is in fact a polynomial of degree N we can represent the function exactly by using an Nth order Legendre basis.
- In the modal interpolation approach the interpolating (basis) functions ϕ represent the various frequencies whereas the expansion coefficients $\widetilde{f}$ are the amplitudes.

Legendre Polynomials: Summary

- The amplitude-frequency space is often called spectral space because we are decomposing the function $f(x)$ into its wave *spectrum* which, when doing so, means that we are evaluating wave *modes* and for this reason we shall refer to this type of approximation as a *modal* approach.
- Before discussing the nodal approach, it should be mentioned that any set of polynomials will work but the best choice is to use orthogonal polynomials such as Legendre, Lobatto, Chebyshev, Fourier, or Jacobi.

Vandermonde Matrix

- We have described interpolation using orthogonal polynomials which is nothing more than a finite series approximation of the function that we would like to interpolate.
- For example, if we take the monomial (power series) approximation of a function, we can write

$$f(x) = \sum_{j=0}^{N} \phi_j(x)\widetilde{f}_j + e_N(x) \equiv \sum_{j=0}^{N} (x - x_0)^j \widetilde{f}_j + e_N(x)$$

where x_0 is the point at which we construct our expansion about.

- Expanding this equation up to a few terms reveals that

$$f(x) = \widetilde{f}_0 + \widetilde{f}_1(x - x_0) + \widetilde{f}_2(x - x_0)^2 + \ldots + \widetilde{f}_N(x - x_0)^N + e_N(x).$$

Vandermonde Matrix

- By sampling the function at $x = x_0$ shows that in fact we have the following conditions on the expansion coefficients

$$\begin{aligned} f(x_0) &= \widetilde{f}_0 \\ f^{(1)}(x_0) &= \widetilde{f}_1 \\ f^{(2)}(x_0) &= 2\widetilde{f}_2 \\ f^{(3)}(x_0) &= 6\widetilde{f}_3 \\ \dots &= \dots \\ f^{(i)}(x_0) &= i!\widetilde{f}_i \end{aligned}$$

which is in fact the Taylor series approximation of $f(x)$ and can be written compactly as

$$f(x) = \sum_{j=0}^{N} \frac{f^{(j)}(x)(x - x_0)^j}{j!} + e_N(x).$$

Vandermonde Matrix

- If we sample the function $f(x)$ at $i = 0, ..., N$ unique x points yields

$$f(x_i) \equiv f_i = \sum_{j=0}^{N} \phi_j(x_i) \widetilde{f}_j$$

which, if we expand it, yields

$$\begin{pmatrix} f_0 \\ f_1 \\ ... \\ f_N \end{pmatrix} = \begin{pmatrix} 1 & x_0 & x_0^2 & ... & x_0^N \\ 1 & x_1 & x_1^2 & ... & x_1^N \\ ... & ... & ... & ... & ... \\ 1 & x_N & x_N^2 & ... & x_N^N \end{pmatrix} \begin{pmatrix} \widetilde{f}_0 \\ \widetilde{f}_1 \\ ... \\ \widetilde{f}_N \end{pmatrix}$$

and can be written in the following matrix-vector form

$$\boldsymbol{f} = \boldsymbol{V} \widetilde{\boldsymbol{f}}$$

where V is known as the standard Vandermonde matrix.

Vandermonde Matrix

- Note that if we would like to map back and forth between modal ($\widetilde{f}$) and nodal space (f) requires the matrix V to be invertible and therefore non-singular (i.e., $\det(V) \neq 0$).
- Therefore, if V is a non-singular matrix then we can write

$$\widetilde{\boldsymbol{f}} = \boldsymbol{V}^{-1}\boldsymbol{f}.$$

- The fact that $\boldsymbol{V}$ must be invertible imposes some restrictions not only on the polynomial functions used to construct the Vandermonde matrix (i.e., monomials versus Jacobi polynomials) but also on the interpolation points used to sample the polynomial functions.
- These interpolation points (or nodal points) are the building blocks of the Lagrange polynomials that we use in a nodal interpolation approach.

Nodal Functions

- Note that the modal interpolation approach and its corresponding Vandermonde matrix implies that there is another set of functions that can be used to approximate $f(x)$ which we write as following

$$f(x) = \sum_{j=0}^{N} L_j(x) f_j \tag{1}$$

where L are the Lagrange polynomials that have the cardinal property

$$L_{ij} = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j \end{cases}$$

which is immediately obvious from Eq. (1) by substituting $x = x_i$.

- The question is how do we construct these functions?

Nodal Functions

- One way of doing this is to use the natural functions of the space, that is, the modal functions.
- Note that the cardinality property of the Lagrange polynomials implies that the following condition must always be satisfied

$$\phi_i(x) = \sum_{j=0}^{N} \phi_i(x_j) L_j(x)$$

where $x_j \; \forall \; j = 0, ..., N$ are any set of interpolation points at which the modal functions, $\phi(x)$, have been sampled.

- Taking $x = x_k$ for some $k = 0, ..., N$ we, in fact see that the cardinality of the Lagrange polynomials yields the desired identity

$$\phi_i(x_k) = \phi_i(x_k)$$

Nodal Functions

- This is the case because

$$L_j(x_k) = \begin{cases} 1 & \text{if} \quad j = k \\ 0 & \text{if} \quad j \neq k \end{cases}$$

- Taking advantage of the fact that $V_{ij} = \phi_i(x_j)$ is in fact the generalized Vandermonde matrix allows us to write

$$V_{ij} L_j(x) = \phi_i(x)$$

and left-multiplying by the inverse of V yields

$$L_i(x) = V_{ij}^{-1} \phi_j(x)$$

Remark

the generalized Vandermonde matrix is different from the standard Vandermonde in that any modal function is used in the general but only the monomials are used in the standard

Nodal Functions

- This now defines the Lagrange polynomials (nodal functions) as functions of the orthogonal modal polynomials.
- Let us define the Lagrange polynomials as follows

$$L_i(x) = \sum_{j=0}^{N} V_{i,j}^{-1} \phi_j(x)$$

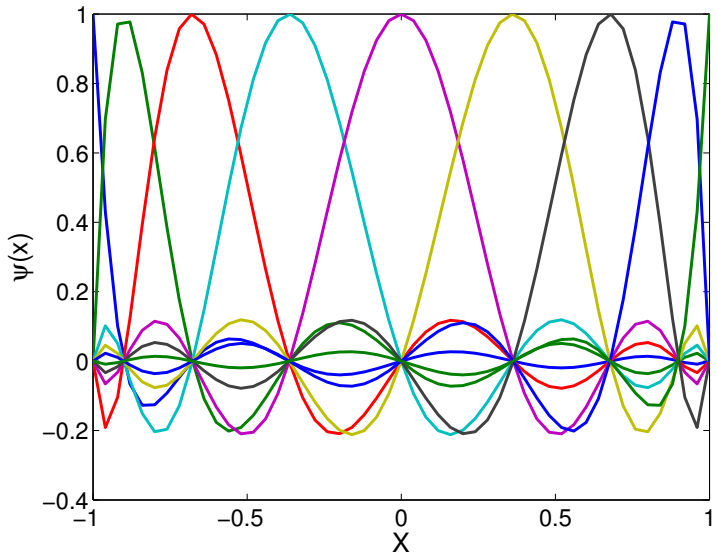
because this form is the one that we shall use to construct the Lebesgue function that is a measure of how good the interpolant is.

Nodal Functions

- In one-dimension, the general definition of nodal functions (Lagrange polynomials) simplifies to

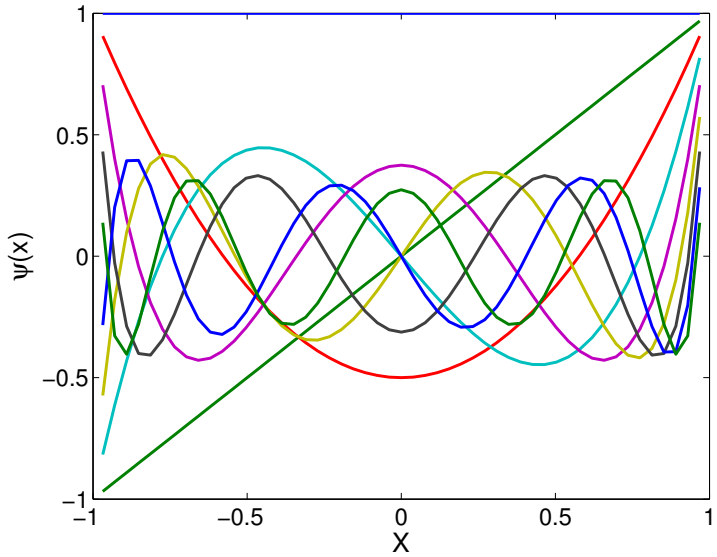
$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{N} \frac{(x - x_j)}{(x_i - x_j)}.$$

- A set of 8th degree Lagrange polynomial (nodal functions) are given as follows



Nodal Functions

- For comparison, here are a set of 8th degree modal functions



Lebesgue Function

- The definition of the Lagrange polynomials that we have used only insures that the functions are cardinal but says nothing about the quality of the interpolation.
- Let us now introduce a measure of interpolation quality.
- For a Lagrange interpolation function a good measure of its interpolation strength is the Lebesgue function and Lebesgue constant (pronounced Lebeck)

$$\Lambda_N(x) = \sum_{i=0}^{N} |L_i(x)|, \quad \Lambda_N = max\left(\sum_{i=0}^{N} |L_i(x)|\right)$$

where the maximum is obtained for the entire domain $x \in D$ where D is the domain of interest (say $x \in [-1, +1]$).

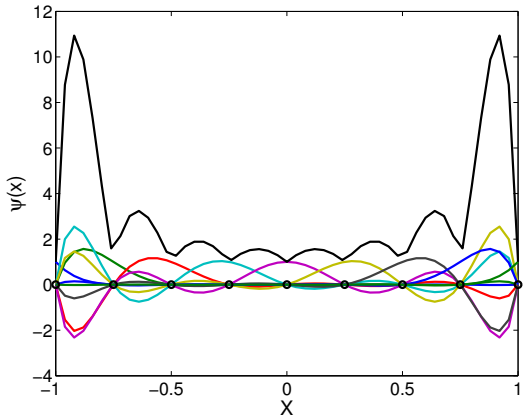
Equi-spaced Interpolation

- Let's consider the following Runge function

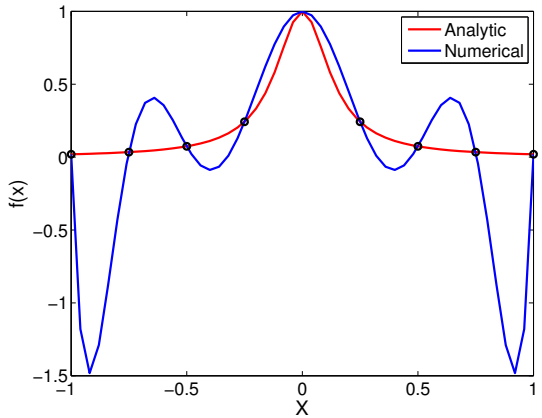
$$f(x) = \frac{1}{1 + 50x^2}$$

defined on the line $x \in [-1, +1]$.

- Using equi-spaced interpolation points with Lagrange polynomials yields the following Lebesgue function (left panel) and interpolation quality (right panel)



a) Lebesgue function



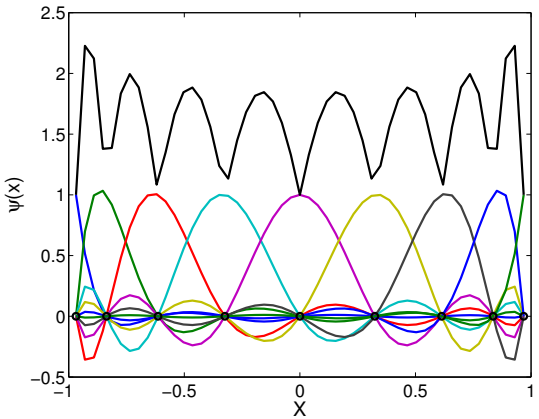
b) Runge function

Legendre Point Interpolation

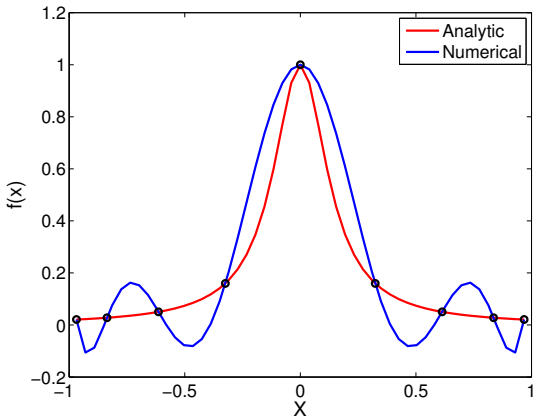
- In contrast, using Legendre polynomials such as $\phi_0^{Leg}(x) = 1, \;\; \phi_1^{Leg}(x) = x,$

$$\phi_N^{Leg}(x) = \frac{2N-1}{N} x \phi_{N-1}^{Leg} - \frac{N-1}{N} \phi_{N-2}^{Leg}(x) \;\; \forall N \geq 2$$

where Nth order points are obtained from roots of (N+1)th order Legendre polynomial, yields the following



a) Lebesgue function



b) Runge function

Lobatto Point Interpolation

- The Lobatto polynomials are derived from the Legendre polynomials as follows

$$\phi_N^{Lob}(x) = (1+x)(1-x)\frac{d}{dx}\phi_{N-1}^{Leg}(x) \equiv (1-x^2)\frac{d}{dx}\phi_{N-1}^{Leg}(x) \quad \forall N \geq 2$$

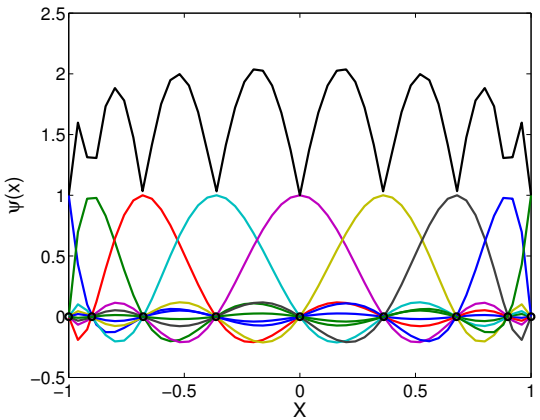
- I.e., the Nth order Lobatto polynomials are nothing more than the (N-1)th derivatives of the Legendre polynomials with the end points $x = \pm 1$ included.

Remark

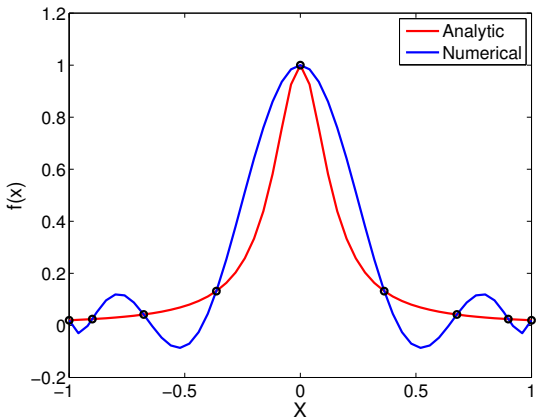
Recall that you need $N+1$ points to obtain a polynomial $\mathcal{O}(N)$.

Lobatto Point Interpolation

- The Nth order Lagrange polynomial is constructed using the roots of the (N+1)th order Lobatto polynomials
- For example, for linear Lagrange polynomial (N=1), we need to find the roots of the 2nd order Lobatto polynomials, $\phi_2^{Lob}(x)$, etc. which yields the following



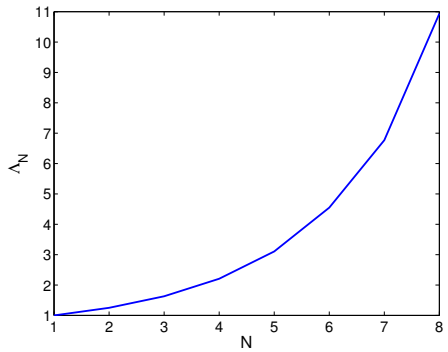
a) Lebesgue function



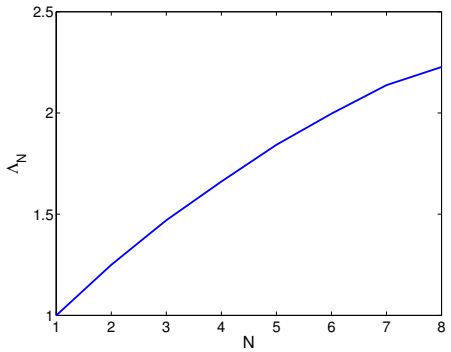
b) Runge function

Comparison of Lebesgue Constants

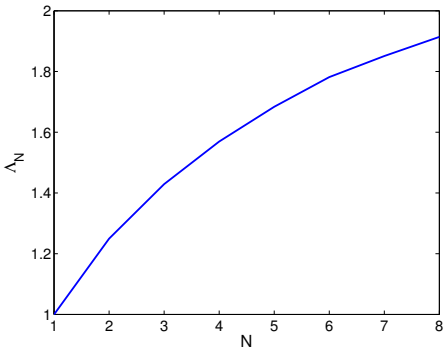
- Let us look at the Lebesgue constants for various Lagrange polynomial orders for different types of points



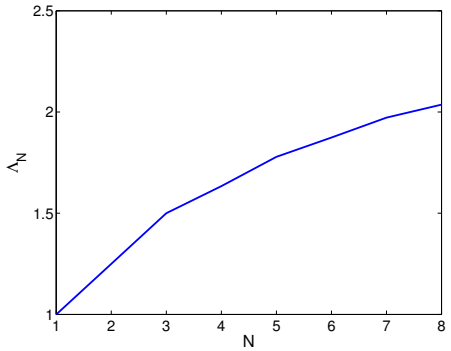
a) Equi-Spaced



b) Legendre



c) Chebyshev



d) Lobatto

Outline

- Overview of existing methods
- Discontinuous Galerkin method
- 1D Interpolation
- 1D Integration

Approximating a Function

- We can approximate a function $f(x)$ as follows

$$f(x) = \sum_{i=0}^{N} a_i x^i + e_N(x)$$

where a_i are the expansion coefficients and $e_N(x)$ the N error remaining after using only N terms in the truncated series.

- Let us rewrite this approximation in the following slightly different form

$$f(x) = \sum_{i=0}^{N} a_i (x - x_0)...(x - x_{i-1}) + e_N(x)$$

which is a power series representation in NBH of x_0.

Approximating a Function

- Let us write the previous eq. as follows

$$f(x) = p_N(x) + e_N(x)$$

where $p_N(x) = \sum_{i=0}^{N} a_i(x - x_0)...(x - x_{i-1})$ is the Nth order polynomial representation of $f(x)$.

- If we want to match the function f at specific points $x_0, x_1, ..., x_N$ then we can figure out unique values of $a_0, a_1, ..., a_N$.
- If we let $x = x_0$ then we see that $f(x_0) = a_0$ and for $x = x_1$ we get

$$f(x_1) = a_0 + a_1(x_1 - x_0)$$

that yields

$$a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

Divided Differences

- Introducing the divided difference notation:

$$\begin{aligned} f[x_i] &= f(x_i) \\ f[x_i, x_{i+1}] &= \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i} \\ f[x_i, x_{i+1}, x_{i+2}] &= \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i} \end{aligned}$$

allows us to write for $x = x_2$

$$f(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1)$$

that, when substituting for a_0 and a_1 yields

$$f(x_2) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1).$$

Divided Differences

- Rearranging gives

$$a_2 = \frac{f(x_2) - f(x_0) - \frac{f(x_1)-f(x_0)}{x_1-x_0}(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)}$$

and, adding and subtracting $f(x_1) - f(x_1)$ in the numerator yields

$$a_2 = \frac{f(x_2) - f(x_1) + f(x_1) - f(x_0) - \frac{f(x_1)-f(x_0)}{x_1-x_0}(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)}.$$

- Upon simplifying this expression we get

$$a_2 = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \equiv f[x_0, x_1, x_2].$$

Polynomial Representation

- Continuing with this approach we get the following expression for the polynomial representation

$$p_N(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + ... + a_N(x - x_0)...(x - x_{N-1})$$

- which can be written more compactly (using Divided differences) as

$$p_N(x) = \sum_{i=0}^{N} f[x_0, ..., x_i] \prod_{j=0}^{i-1} (x - x_j)$$

Theorem

- *For $f \in C^N[a, b]$ and a set of distinct points $(x_0, ..., x_N) \in [a, b]$ there exists $\xi \in (a, b)$ such that $f[x_0, x_1, ..., x_N] = \frac{f^{(N)}(\xi)}{N!}$.*

Polynomial Representation

- The importance of this theorem is that it now tells us that the polynomial approximation now takes the following form

$$f(x) = p_N(x) + \frac{f^{(N)}(\xi)}{(N)!} \prod_{i=0}^{N} (x - x_i)$$

where the last term is in fact the error that clearly vanishes when x is one of the points x_i for $i = 0, ..., N$ since we imposed that the approximation satisfy the true function at these points.

Polynomial Representation

- Let us rewrite the previous eq. in the following form:

$$f(x) = p_N(x) + \phi_N(x) f[x_0, x_1, ..., x_N, x]$$

that allows us to see that if $f(x)$ is of degree M and $p_N(x)$ is of degree N (where $M > N$), then $\phi(x)$ is of degree N and $f[x_0, ..., x_N, x]$ is of degree $M - N$

- note that

$$\phi(x) = \prod_{i=0}^{N} (x - x_i)$$

is the generating polynomial.

Numerical Integration

- Let us suppose that we wish to integrate the following equation

$$f(x) = p_N(x) + \phi_N(x) f[x_0, x_1, ..., x_N, x]$$

in the interval $[a, b]$.

- We can then write

$$I = \int_a^b f(x)\, dx = \int_a^b p_N(x)\, dx + \int_a^b e_N(x)\, dx$$

where

$$e_N(x) = \phi_N(x) f[x_0, x_1, ..., x_N, x]$$

is the error term and if we wish to satisfy the integral exactly requires that

$$\int_a^b e_N(x)\, dx = 0.$$

Numerical Integration

- Note that this condition is satisfied provided that ϕ_N is of degree N and $f[x_0, x_1, ..., x_N, x]$ is of degree less than or equal to N-1 since, for families of orthogonal polynomials, all Nth degree polynomials are orthogonal to all polynomials of degree less than or equal to N-1

Remark

you can see this by picturing how Gram-Schmidt orthogonalization builds families of orthogonal polynomials

- Note that the product $\phi_N f[x_0, x_1, ..., x_N, x]$ is a polynomial of degree $2N - 1$.

Numerical Integration

- This brief analysis shows that one can achieve order $2N-1$ integration accuracy by choosing orthogonal polynomials of degree N for the same family for ϕ_N and $f[x_0, ..., x_N, x]$, where the sampling points for the numerical integration are the roots of orthogonal polynomials.
- Note, however, that we have not mentioned how to compute the quadrature weights.

Numerical Integration: Summary

- In sum, if we use Legendre quadrature then we have $N+1$ roots and $N+1$ weights which leads to DOF$=2N+2$ and thereby allows us to create an $\mathcal{O}(2N+1)$ polynomials (DOF-1).
- If we use Lobatto quadrature we have $N-1$ free roots (2 fixed endpoints) but still have $N+1$ weights for a total of DOF$=2N$ producing a polynomial of order $2N-1$.

Quadrature Roots

- Since we know that the sampling points for numerical integration must be the roots of orthogonal polynomials, we must now describe how to compute them.
- Let $\phi_N(x)$ be an Nth order orthogonal polynomial.
- We can write the Newton's method solution to this nonlinear problem as

$$\phi_N\left(x^{k+1}\right) = \phi_N\left(x^k\right) + \left(x^{k+1} - x^k\right)\frac{d\phi_N\left(x^k\right)}{dx}.$$

  where x^k is the kth approximation to the roots.
- This equation becomes zero when x^k is equal to the root of ϕ_N.

Quadrature Roots

- Therefore, we iterate to find the roots as follows

$$x^{k+1} = x^k - \frac{\phi_N\left(x^k\right)}{\frac{d\phi_N\left(x^k\right)}{dx}}.$$

- Recall that for Newton's method, we need a good guess in order to converge to the proper solution.
- We use the Chebyshev roots as the initial guess that are quite near to the Legendre and Lobatto roots and are obtained in closed form as follows

$$x_i^0 = \cos\left(\pi \frac{2i+1}{2N+2}\right) \ \forall i = 0, \ldots, N.$$

Quadrature Weights

- Let us write the numerical integration (quadrature) problem as follows

$$I = \int_{-1}^{+1} f(x)\, dx \tag{2}$$

- where, without loss of generality, we have assumed that the interval of integration is defined on the canonical one-dimensional element $x \in [-1, +1]$.
- Next, let us approximate the function $f(x)$ by an Nth order Lagrange polynomial interpolant

$$f(x) = \sum_{j=0}^{N} \psi_j(x) f_j \tag{3}$$

- where ψ_j are the nodal interpolation functions (i.e., Lagrange polynomials) and $f_j = f(x_j)$ are the values of $f(x)$ at specific points $x_j \in [-1, +1]$.

Quadrature Weights

- Substituting Eq. (3) into (2) yields

$$I = \int_{-1}^{+1} \left(\sum_{j=0}^{N} \psi_j(x) f_j \right) \, dx. \tag{4}$$

- Using a quadrature formula, we can represent Eq. (4) as follows

$$\int_{-1}^{+1} \left(\sum_{j=0}^{N} \psi_j(x) f_j \right) \, dx = \sum_{i=0}^{N} w_i \left(\sum_{j=0}^{N} \psi_j(x_i) f_j \right) \tag{5}$$

- where w_i are the weights corresponding to the quadrature points $x_i \, \forall i = 0, \ldots, N$ which are the roots of the Nth order generating polynomials ϕ_N (i.e., the modal functions).

Quadrature Weights

- Reordering the summation and integral signs in Eq. (5) yields

$$\sum_{j=0}^{N} \left(\int_{-1}^{+1} \psi_j(x)\, dx \right) f_j = \sum_{j=0}^{N} \left(\sum_{i=0}^{N} w_i \psi_j(x_i) \right) f_j. \qquad (6)$$

- From Eq. (6), it becomes evident that we can solve for the quadrature weights from the linear matrix problem

$$w_i = \int_{-1}^{+1} \psi_i(x)\, dx \qquad (7)$$

- where we have used the cardinality property of Lagrange polynomials to cancel the term $\psi_j(x_i)$ from the right-hand-side of Eq. (6).
- For certain orthogonal polynomials, the quadrature weights can be shown to have very simple closed form representations.

Summary of DG Lecture 1

- We covered the background (theoretical) for DG methods.
- We saw that DG uses an integral (weak) form.
- We saw that DG requires good interpolation and good integration.
- One can construct a modal DG approach (using orthogonal polynomials) or a nodal DG approach (using Lagrange polynomials).
- In the next lecture (Lecture 2) we will see how to apply these ideas to the 1D wave and shallow water equations.