The Fundamentals of Grid Generation

-Knupp & Steinberg-

1.1 : Preliminaries

- In many applications we can transform the physical region into a square in 2D or a cube in 3D in such a way that the boundary of the square or cube corresponds to the boundary of the physical region
  ↳ Square or Cube → "logical region"
  ↳ the transformation gives rise to a boundary conforming coordinate system

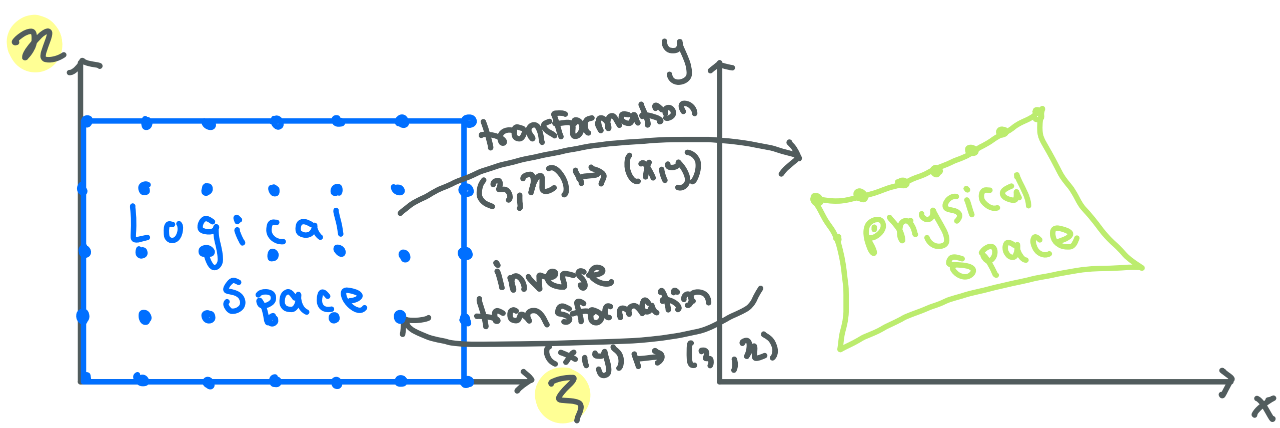
- The coordinate lines in this boundary conforming coordinate system are given by the images of the uniform coordinate lines in the logical region.

- In these coordinate systems we can make accurate implementations of numerical BC's

- The Jacobian of the transformation is required to be nonzero so that the transformation has an inverse. This means we can map from the logical space back to the physical space.

- Since grids are chosen in the logical space & then mapped to the physical space, we view the transformation as a mapping from the logical to physical space (we focus on the inverse transformation)

visualize



- If the Jacobian of the transformation is ever zero, the transformation fails to preserve the essential physical & mathematical properties of the hosted eqns.

folded transformations → have zero jacobian → BAD

- The error of the approximations of the hosted eqns (physical problem) depends on the derivatives of the soln of the hosted eqn, the grid spacing, and the departure from grid orthogonality

- For a given grid spacing, smooth & orthogonal grids usually result in the smallest error in simple problems. Therefore we want smooth grids

Smooth grids → spacing varies smoothly & angles btwn grid lines dont get too small

Solution- adapted grids

$\boxed{\text{Finer grid in regions where the physical solution varies rapidly to reduce the error in the numerical soln}}$

our focus!

1.2 : Mappings & Invertibility

domain of mapping → logical space (ξ, η, ζ)

range of mapping → physical space (x, y, z)

- These spaces are limited to subsets of the Euclidean spaces E^n $(n = 1, 2, \text{ or } 3)$
- Sometimes its better to use coordinates w/ indices

  logical space : $\vec{\xi} = (\xi_1, \xi_2, \ldots, \xi_k)$

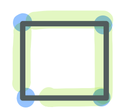
  Physical Space : $\vec{x} = (x_1, x_2, \ldots, x_n)$

Logical Space

$U_1 = \{ \xi \in E^1 ; 0 \le \xi \le 1 \}$ = unit interval → 2 points

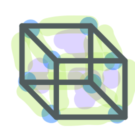
$U_2 = \{ (\xi, \eta) \in E^2 ; 0 \le \xi, \eta \le 1 \}$ = unit square

4 segments &
4 points



$U_3 = \{ (\xi, \eta, \zeta) \in E^3 ; 0 \le \xi, \eta, \zeta \le 1 \}$ = unit cube

6 faces,
12 segments, 8 points



Physical Space

- In 2D its nececessary to place grids on regions (2D) & curves (1D)
- In 1D only intervals need grids

- Objects in the physical space have two important parameters

(a) dimension of the object (k)

(b) dimension of physical space (n)

A k dimensional object in an n-dimensional physical space is denoted Ω^n_k & is assumed bounded.

Mapping

- Transformations from the logical to physical objects give a system of general coordinates on the physical object. Such maps have a parameter k for the dimension of the object & a dimension n for the dimension of the physical space (assume $0 < n \leq 3$, $0 < k \leq n$)

$$x^n_k : U_k \to \Omega^{(n)}_k \qquad (1.1)$$

- In general, such maps are written as

$$x = x(\xi) \qquad (1.2)$$

- A transformation from logical space → physical space produces a natural grid in physical space. This grid depends on the parametrization of the transformation/map. Thus grid generation can be thought of as finding useful parametrizations of maps

Examples

interval ↦ interval

map : X_1^1

coordinates : $x = x(\xi)$, $y = y(\xi)$

interval ↦ curve (2D)

map : X_1^2

coordinates : $x = x(\xi)$, $y = y(\xi)$

interval ↦ curve (3D)

map : X_1^3

coordinates : $x = x(\xi)$, $y = y(\xi)$, $z = z(\xi)$

square $\mapsto$ region (2D)

$$\text{map: } X_2^2$$
$$\text{coordinates: } x = x(\xi, \eta), \quad y = y(\xi, \eta)$$

square $\mapsto$ surface (3D)

$$\text{map: } X_2^3$$
$$\text{coordinates: } x = x(\xi, \eta), \quad y = y(\xi, \eta), \quad z = z(\xi, \eta)$$

cube $\mapsto$ volume

$$\text{map: } X_3^3$$
$$\text{coordinates: } x = x(\xi, \eta, \zeta), \quad y = y(\xi, \eta, \zeta), \quad z = (\xi, \eta, \zeta)$$

- Before a grid can be generated, a physical object must be specified mathematically by specifying its boundary. This boundary can be given parametrically, numerically, or implicitly.

ex: Boundary of unit disk

→ parametrically: $x = \cos(\theta)$, $y = \sin(\theta)$, $\theta \in [0, 2\pi]$

→ implicitly: $x^2 + y^2 = 1$

- numerically specified boundaries can be converted to parametrically specified boundaries via interpolation

- It is required that the mapping maps the boundary of the logical region (∂U_k) to the boundary of the physical region ($\partial \Omega_k^n$). In general we start w/ this mapping: $\partial X_k^n : \partial U_k \to \partial \Omega_k^n$, and extend it to the interior of U_k

- If an object is given implicitly, a parametric representation of the boundary must be found

- Regions assumed simply connected (no holes)
(can be extended too)

The Basic Grid Generating Problem

The MAP

If the boundary of the physical object is given by a nonsingular parametric map: $\partial X_k^n : \partial U_k \to \partial \Omega_k^n$ (1.7), then we extend this map to, $X_k^n : U_k \to \Omega_k^n$ (1.8), which maps from the interior of the logical space to the interior of the physical object & is a boundary conforming map

Bad Things

Spill over or folding → Point in logical space is mapped to a point outside of the physical object

not a bijection → two or more points in the logical space are mapped to by the same point on the physical object

↳ to every $\xi \in U_k$ we require that there corresponds a unique point $x \in \Omega_k^n$. And to every point $x \in \Omega_k^n$ we require that there corresponds a unique point $\xi \in U_k$.

Smoothness

- The map X_n^k is generally required to be smooth → each of the coordinate functions $x_i(\xi)$ must be continuous & have continuous derivatives as functions of ξ_i on U_k → map must be smooth on U_k & ∂U_k → corners of logical space must map to corners of physical object

$C^j \rightarrow$ space of maps whose coordinate functions are in C^j

$\rightarrow$ have j continuous derivatives

Assumptions

- It will be assumed that $X_k^n \in C^1$
- If $X_k^n : U_k \rightarrow \Omega_k^n$ is a bijection & $X_k^n \in C^1$ then X_k^n is a diffeomorphism
- The boundary mapping ∂X_k^n is assumed to be a diffeomorphism

↳ sometimes this can be relaxed (finite # of points)

↳ sometimes it must be strengthened

General Grid Generation Problem

Extend a given boundary diffeomorphism ∂X_k^n to a diffeomorphism X_k^n of U_k to Ω_k^n

The Jacobian J

- Since it is assumed $X_k^n \in C^1 \rightarrow \frac{\partial x_i}{\partial \xi_j}$ are defined

$$\mathcal{J} = \nabla X_k^n$$

$$\rightarrow J_{ij} = \frac{\partial x_i}{\partial \xi_j} \qquad \begin{array}{l} i = 1, \dots, n \\ j = 1, \dots, k \end{array}$$

- In general, $\mathcal{J}$ is not square
- When $\mathcal{J}$ is square, det($\mathcal{J}$) exists and is referred to as the Jacobian J of the map X_k^k

$$J = \det(\mathcal{J})$$

Inverse Mapping Thm

Assume $X_k^n \in C^1$. Then X_k^n is Locally (all points near a given point) one-to-one at $\mathfrak{z}$ in the interior of U_k $\Leftrightarrow$ the rank of $\mathcal{J}$ is maximal (equals k) at $\mathfrak{z}$ (is a nonsingular map)

- If $n=k$ (object not a curve or surface) the jacobian matrix is square. Square matrices have maximal rank $\Leftrightarrow$ determinants $\neq 0$

Corollary

Assume $X_n^n \in C^1$
X_n^n is locally one-to-one near $\mathfrak{z}$ $\Leftrightarrow$ $J(\mathfrak{z}) \neq 0$

Ch. 5 → Classical Planar Grid Generation

Elliptic Methods → generate grids by solving bvps for elliptic PDE's.

5.2 → The Planar Grid Problem

Given a simply-connected region $\Omega \subset \mathbb{R}^2$ in physical space, find a mapping from the unit square U_2 in the logical space E^2 to the region Ω

The physical region is specified by giving its boundary. This may be done by giving a set of four parametric maps

$x_b(s)$, $x_t(s)$, $0 \leq s \leq 1$ (bottom, top)
$x_l(s)$, $x_r(s)$, $0 \leq s \leq 1$ (left, right)

logical domain boundaries

while the desired map is written in components as

$$x = x(\xi,\eta) \;,\; y = y(\xi,\eta)$$

$$(\xi,\eta) \mapsto (x,y)$$

logical $\mapsto$ physical

Variations

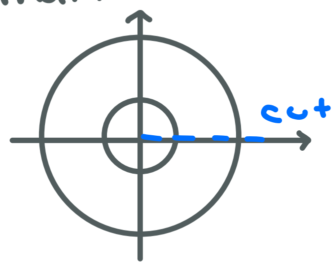
- The given boundary may be re-parametrized before we extend to the interior. Thus if $s = s(r)$ w/ $0 \le r \le 1$, then the composite function $x_b = x_b(s(r)) = \hat{x}_b(r)$ is a re-parametrization of the bottom boundary of Ω. Re-parametrization is an important technique since the original boundary parametrization is frequently inconvient

- We can reduce the number of boundaries that are to be fit

  ex: Only three boundaries are need in some approaches to orthogonal grid generation and only one is needed for hyperbolic grid generation. The other boundaries are then free & out of user control

- The requirement that a region must be simply connected may sometimes be relaxed by introducing cuts into the physical domain

ex: Annulus



- regions not topologically equivalent to a square may require modification of the logical domain

The Discrete Transformation

- The transformations are computed via discrete approx's
- Let M and N be positive integers
- Define the logical grid points (ξ_j, η_j) by

logical Domain

$$\xi_i = \frac{i}{M} \qquad 0 \le i \le M \rightarrow M \ \xi_i \text{ points}$$

$$\eta_j = \frac{j}{N} \qquad 0 \le j \le N \rightarrow N \ \eta_j \text{ points}$$

($M \times N$ points, uniform spacing)

- let $\Delta \xi = \frac{1 \rightarrow b-a}{M}$ and $\Delta \eta = \frac{1 \rightarrow b-a}{N}$ (on $[0,1] \times [0,1]$)

- The transformation $x = x(\xi, \eta)$ and $y = y(\xi, \eta)$ carries logical-space grid $\rightarrow$ physical-space grid $X_{i,j} = (x_{i,j}, y_{i,j})$ where

Mapping

$$\boxed{\begin{gathered} x_{i,j} = x(\xi_i, \eta_j), \quad y_{i,j} = y(\xi_i, \eta_j) \\ 0 \le i \le M \\ 0 \le j \le N \end{gathered}}$$

5.4 → Elliptic Grid Generation

Elliptic Grid Generators → solve systems of elliptic pde's to generate grids

Advantages

- interior grid is very smooth, even for nonsmooth boundary data. This grid smoothness allows us to achieve low truncation error in physical problem (finite diff)
- One can specify all four bounderies of the domain
- Interior grid is relatively insensitive to the boundary parametrization → sols exist for a wide variety of parametrizations & change only gradually as boundary data changes.

Disadvantages

- harder to solve → slower
- Harder to control the interior grid

Simplest Generator (Length)

Problem

We require each component of the map to satisfy Laplace's Eqn

$$\nabla^2 x = \nabla_\xi^2 x = x_{\xi\xi} + x_{\eta\eta} = 0$$

$$\nabla^2 y = \nabla_\xi^2 y = y_{\xi\xi} + y_{\eta\eta} = 0$$

with the boundary data

bottom ($\eta = 0$) $x(\xi, 0) = x_b(\xi)$, $x(\xi, 1) = x_t(\xi)$ top ($\eta = 1$)

left ($\xi = 0$) $x(0, \eta) = x_l(\eta)$, $x(1, \eta) = x_r(\eta)$ right ($\xi = 1$)

(Solve problem on (ξ, η) to obtain (x, y) which define the physical space)

Analysis

- Eqns for x and y are uncoupled, linear, and formulated on a square logical domain. The PDE's are Laplace Eqns.

- The BC's are Dirichlet
- The BVP is called the Dirichlet BVP for Laplace's eqn
- Such problems have unique soln that is infinitely differentiable (in C^∞) in the interior of U_2 provided the boundary maps are continuous. This is not important for grid generation b/c if the boundary maps are not continuous, the physical region is not defined.

- Are the maps bijections (is the Jacobian of the map nonzero)? → NO → the maps typically fold for non-convex regions

discretisation

- The second derivatives are discretized w/ standard central differences

$$\frac{1}{\Delta \xi^2}\left(x_{i-1,j} - 2x_{i,j} + x_{i+1,j}\right) + \frac{1}{\Delta \eta^2}\left(x_{i,j-1} - 2x_{i,j} + x_{i,j+1}\right) = 0$$

for $1 \leq i \leq M-1, \quad 1 \leq j \leq N-1$

- The numerical BC's are

$$x_{i,0} = x_b(\xi_i) \qquad x_{i,N} = x_t(\xi_i) \qquad 0 \leq i \leq M$$
$$x_{0,j} = x_\ell(\eta_j) \qquad x_{M,N} = x_r(\eta_j) \qquad 0 \leq j \leq N$$

The Winslow or Smoothness Generator

→ deals w/ folding problem on non-convex domains that occur for length generator

- The method arose from the need for an elliptic generator that would produce un-folded grids (transformation w/ positive jacobian)

It requires the components of the inverse transformation

$$\xi = \xi(x,y), \quad \eta = \eta(x,y)$$

to solve Laplace's Equation

$$\nabla^2 \xi = \nabla_x^2 \xi = \xi_{xx} + \xi_{yy} = 0$$

$$\nabla^2 \eta = \nabla_x^2 \eta = \eta_{xx} + \eta_{yy} = 0$$

on a convex local domain.

- Dirichlet BC's are given by the inverses of the boundary maps

- By requiring that the inverse map be harmonic (instead of map itself in length generator), we can show that the map is one-to-one

- In fact, Winslow mapping is a diffeomorphism provided the boundary map is a homeomorphism
  - ↳ The continuum soln to the Winslow Eqns results in an unfolded transformation on the interior of the physical domain

Numerical Solution

- Hard to solve directly on an arbitrary domain so we transform them to the logical space.

$$g_{22} x_{\xi\xi} - 2 g_{12} x_{\xi\eta} + g_{11} x_{\eta\eta} = 0$$

$$g_{22} y_{\xi\xi} - 2 g_{12} y_{\xi\eta} + g_{11} y_{\eta\eta} = 0$$

where
$$g_{11} = x_\xi^2 + y_\xi^2 \neq 0$$
$$g_{12} = x_\xi x_\eta + y_\xi y_\eta \neq 0$$
$$g_{22} = x_\eta^2 + y_\eta^2 \neq 0$$

$g_{11} \to$ length^2 of a tangent vector to a ξ-coordinate line
$g_{22} \to$ length^2 of a tangent vector to a η-coordinate line
$g_{12} \to$ inner product of the two tangent vectors

$$\Rightarrow \qquad g_{i,j} = X_{\xi_i} \cdot X_{\xi_j}$$

Quasi-linear Winslow Operator

- The eqns are neither linear nor coupled. We couple them through metric-matrix coeff's which depend on both x and y. These coeff's make the eqns quasi-linear.

- Q_w = Quasi-linear Winslow Operator

$$= g_{22}\frac{\partial^2}{\partial \xi^2} - 2g_{12}\frac{\partial^2}{\partial \xi \partial \eta} + g_{11}\frac{\partial^2}{\partial \eta^2}$$

$$\Rightarrow \qquad Q_w \vec{x} = \vec{0}$$