GAMES 301 Lab 2

Komeiji Green

November 7, 2022

Contents

1 作业内容

Analytic Eigensystems for Isotropic Distortion Energies

利用论文 [1] 中所描述的 Symmetric Dirichlet 能量的解析求法，实现了用于初始无翻转离散网格参数化的 Analytic Projected Newton 方法。

实验：观察 Newton 方法二阶收敛性

本作业还实现了经典的一阶方法 SLIM[2]，并将其与上述二阶方法进行对比实验，可以观察到 Newton 方法所具备的二阶收敛性。

压力测试：Hilbert Curve

综合 SLIM 与 Analytic Proj. Newton 方法，在 Hilbert Curve 模型上进行算法压力测试。

2 如何运行

在 MATLAB R2021b 或更新版本上直接运行 demo-xxx 脚本即可, 注意要将文件夹 io, mesh, src, utils 加入路径.

3 算法介绍

本节主要涉及一些算法的原理性介绍以及一些我个人的理解。3.1 概述 Proj. Method 方法，3.2 和 3.3 介绍其部分内容的细节。

3.1 Projected Newton Method

Newton 法的主要思想是，利用目标函数 f 在 $\mathbf{x}$ 处的梯度与 Hessian 矩阵，来得到目标函标函数在 $\mathbf{x}$ 附近的一个二次近似。

在 Line Search 的框架下，Newton 法通过最小化二次近似函数，来获得一个线搜索的方向 $\mathbf{d}$，并沿此方向寻找到下一个可行解 $\mathbf{x}+\alpha\,\mathbf{d}$，牛顿法具有二次收敛性，其收敛所需的迭代次数通常远小于梯度下降等一阶方法。

然而，f 的 Hessian 矩阵有可能不是正定的，这可能导致二次近似的极值无法求解，或者线搜索的方向并不是 f 的下降方向。因此许多方法尝试修改 Hessian 矩阵，将其投影到一个与之相差足够小的半正定矩阵上，也就是本文的 Proj. Newton 方法。

在参数化中，Hessian 的结构与曲面相关，我们可以将其分解为：

$$\mathbf{H} = \mathbf{D}^\top * \mathbf{H_J} * \mathbf{D},$$

其中 $\mathbf{D} \in \mathbb{R}^{4f \times 2n}$ 是 Deform Gradient 算子，$\mathbf{H_J} \in \mathbb{R}^{4f \times 4f}$ 是一个分块对角矩阵，它的每一块代表一个面元，表示能量相对于该面元的 Jacobian 的 Hessian 矩阵。算子 D 将能量相对于点坐标的 Hessian 映射为相对于面元上 Jacobian 的 Hessian。

在这种分解下，只需将 $\mathbf{H_J}$ 的每一块都投影为半正定矩阵，也就完成了 $\mathbf{H}$ 的半正定投影。

ALGORITHM 1: Projected Newton Pseudocode. Our approach allows `Eval_Energy_EigenSystem`($\mathbf{U}, \Sigma, \mathbf{V}$) to be implemented in closed-form.

Function `Projected_Newton_Solver`($\mathbf{x}_0$)
  for $i \leftarrow 0$ ***to*** n **do**
    $\mathbf{b}_i \leftarrow \nabla\Psi(\mathbf{x}_i)$ // Equation (5a)
    if $\|\mathbf{b}_i\|_\infty \leq 10^{-4}$ **then**
      return $\mathbf{x}_i$
    end
    $\mathbf{H}_i \leftarrow$ `Project_Hessian`($\mathbf{x}_i$)
    $\mathbf{d}_i \leftarrow -\mathbf{H}_i^{-1}\mathbf{b}_i$
    $\alpha_i \leftarrow$ `Line_Search`($\mathbf{x}_i, \mathbf{d}_i$)
    $\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \alpha_i \mathbf{d}_i$
  end
  return $\mathbf{x}_{n+1}$
Function `Project_Hessian`($\mathbf{x}$)
  $\mathbf{H} \leftarrow \mathbf{0}$
  for *every quadrature point* q **do**
    $\mathbf{F} \leftarrow$ `Compute_Deformation_Gradient`($q, \mathbf{x}$)
    $\mathbf{f} \leftarrow \mathrm{vec}(\mathbf{F})$
    $\{\mathbf{U}, \Sigma, \mathbf{V}\} \leftarrow$ `Compute_SVD`($\mathbf{F}$)
    $\{\lambda_i, \mathbf{e}_i\} \leftarrow$ `Eval_Energy_EigenSystem`($\mathbf{U}, \Sigma, \mathbf{V}$)
    $\mathbf{H}_q \leftarrow \sum_i \max(\lambda_i, 0)\, \mathbf{e}_i \mathbf{e}_i^\top$
    $\mathbf{H} \leftarrow \mathbf{H} + |q| \left(\partial \mathbf{f}/\partial \mathbf{x}\right)^\top \mathbf{H}_q \left(\partial \mathbf{f}/\partial \mathbf{x}\right)$ // Equation (5b)
  end
  return $\mathbf{H}$

Figure 1: Projected Newton Method 主体流程

3.2 Line Search

3.2.1 无翻转条件

为了保证参数化结果是无翻转的，Line Search 的步长不能使任何一个三角形发生翻转。为此，对每一个三角形，我们寻找使其发生退化的最小的 α，也就是该三角形发生翻转的临界值，取所有临界值的最小值作为 Line Search 步长的上界，即可保证无翻转的要求。

注意，SD 能量的作用是使算法**更倾向于**在无翻转的方向上进行 Line Search，但仅仅依赖 SD 函数无法严格保证无翻转。虽然 SD 能量在三角形发生退化时会趋于无穷，但是 Line Search 的过程只考虑了搜索方向上的一些离散点，难免跳过 SD 函数的障碍，因此要通过上述方法来硬性地限制翻转的发生。

3.2.2 Wolfe 条件

可以通过如下算法来寻找满足 Wolfe 条件的步长，以使得能量函数充分下降。其中，我们设初值 $\alpha_{min} = 0$，α_{max} 小于发生翻转的临界值。

输入的 ϕ 是一个关于 α 的一维函数，在算法的过程中，只需在部分点上对其求值。

Algorithm 2: 寻找 Wolfe 条件的二分算法

```
Input: φ(α) = x + α d, α_min, α_max
Output: α_*
for j ← 1 : N do
    α_j = (α_min + α_max) / 2;
    if φ(α_j) > φ(0) + c_1 α_j φ'(0) then
        α_max ← α_j;
    else
        if |φ'(α_j)| < c_2 |φ'(0)| then
            return α_j;
        else if φ'(α_j) < 0 then
            α_min ← α_j;                    /* Make it bigger */
        else
            α_max ← α_j;                    /* Make it smaller */
        end
    end
end
```

根据 Numerical Optimization 第三章中的引理 3.1 [3]，当算法 2 的迭代次数 N 充分大时，我们一定可以找到一个 α 满足 Strong Wolfe 条件，即：

$$\phi(\alpha) < \phi(0) + c_1 * \phi'(0) * \alpha; \tag{1}$$

$$|\,\phi'(\alpha)\,| < c_2 * |\,\phi'(0)\,| \tag{2}$$

超参数 c_1, c_2 通常要满足：$0 < c_1 < c_2 < 1$，作业中我通常选取 $c_1 = 10^{-5}, c_2 = 0.99$

3.3 Analytic Eigen System

论文作者 Smith 在 18、19 年的 TOG 文章上似乎都没有明确指出一些性质，我认为它们对于理解整个文章还是很有帮助的：

Lemma 3.1. *设* $\Psi : \mathbb{R}^{n\times n} \to \mathbb{R}$ *是某个旋转不变函数，即对任意正交矩阵* $\mathbf{U}, \mathbf{V}$*，有* $\Psi(\mathbf{U}\mathbf{A}\mathbf{V}^\top) = \Psi(\mathbf{A})$*. 考虑 svd 分解* $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$*，若矩阵* Σ *是 4-th tensor* $\nabla^2\Psi(\mathbf{S})$ *的 eigen-matrix，则* $\mathbf{U}\Sigma\mathbf{V}^\top$ *也是* $\nabla^2\Psi(\mathbf{A})$ *的 eigen-matrix.*

这条引理告诉我们，想要求解 $\nabla^2\Psi(\mathbf{A})$ 的 eigen-system，只需求解 $\nabla^2\Psi(\mathbf{S})$ 的 eigen-system，而 $\mathbf{S}$ 是一个由奇异值组成的对角矩阵，此时的 Hessian 的特征向量较易解出.

我们先证明引理 3.1，再用它给出论文 [1] 4.2 节中 Eigensystem of I_1 的推导。

Proof. 考虑 $\nabla\Psi$ 的方向导数：

$$\mathrm{d}[\nabla\Psi]_{\mathbf{A}}(\mathbf{X}) := \lim_{t\to 0} \frac{\nabla\Psi(\mathbf{A}+t\mathbf{X}) - \nabla\Psi(\mathbf{A})}{t}$$

考虑梯度与方向导数的关系，可以证明：$\nabla^2\Psi(A) : \mathbf{X} = \mathrm{d}[\nabla\Psi]_{\mathbf{A}}(\mathbf{X})$.

根据 SLIM [2] 中的 Lemma 3.1，可以推出，$\nabla\Psi(\mathbf{U}\mathbf{A}\mathbf{V}^\top) = \mathbf{U}\nabla\Psi(\mathbf{A})\mathbf{V}^\top$，于是：

$$\begin{aligned}
\nabla^2\Psi(\mathbf{A}) : (\mathbf{U}\Sigma\mathbf{V}^\top) &= \mathrm{d}[\nabla\Psi]_{\mathbf{A}}(\mathbf{U}\Sigma\mathbf{V}^\top) \\
&= \lim_{t\to 0} \frac{\nabla\Psi(\mathbf{A}+t\mathbf{U}\Sigma\mathbf{V}^\top) - \nabla\Psi(\mathbf{A})}{t} \\
&= \lim_{t\to 0} \frac{\mathbf{U}\left[\nabla\Psi(\mathbf{S}+t\Sigma) - \nabla\Psi(\mathbf{S})\right]\mathbf{V}^\top}{t} \\
&= \mathbf{U} * \mathrm{d}[\nabla\Psi]_{\mathbf{S}}(\Sigma) * \mathbf{V}^\top \\
&= \mathbf{U} * [\nabla^2\Psi(\mathbf{S}) : \Sigma] * \mathbf{V}^\top \\
&= \lambda\mathbf{U}\Sigma\mathbf{V}^\top
\end{aligned}$$

这样，$\mathbf{U}\Sigma\mathbf{V}^\top$ 不仅是 eigen-matrix，还与 Σ 具有相同的特征值。 □

把 Hessian 的特征向量与方向导数联系起来，后续的证明也会沿用这种思路。

Eigensystem of I_1

对于矩阵 $\mathbf{J}$，记它的奇异值的和为 I_1，根据 [1] 中的结论，$\nabla_{\mathbf{J}} I_1(\mathbf{J}) = \mathbf{U}\mathbf{V}^\top$，其中 $\mathbf{U}, \mathbf{V}$ 是 $\mathbf{J}$ 做奇异值分解后得到的旋转矩阵。注意到 ∇I_1 始终是一个正交矩阵。

我们知道，对于一个随时间 t 而连续变化的正交矩阵 $\mathbf{P_t}$，有 $\mathbf{P_t}' = \Omega_t * \mathbf{P_t}$，其中 Ω_t 是一个反对称矩阵。在二维情形下，它表示逆时针旋转 90 度，在三维情形，它表示一个叉乘。

那么考虑 I_1 在 $\mathbf{S}$ 处沿 eigen-matrix Σ 的方向导数，必有：

$$\mathrm{d}[\nabla I_1]_S(\Sigma) = \Omega * \mathbf{I} = \lambda\Sigma,$$

其中 I 是单位矩阵，Ω 是一个反对称矩阵。此式说明特征矩阵 Σ 也是反对称的。

In 2D

对于 2 维情形，由上述结论可以立刻推出，I_1 的 Hessian tensor 有且仅有一个特征矩阵：

$$\Sigma_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},$$

对于特征值的求解，可以将 Σ_1 代入方向导数，并利用公式 $I_1 = \sqrt{||\mathbf{J}||_F^2 + 2\det\mathbf{J}}$ 算出，最终结果 $\lambda = \frac{2}{\sigma_1+\sigma_2}$.

In k-D

对于 k 维情形，很容易将问题化归为 2 维情形。我们考虑最基本的反对称矩阵，比如：将 Σ_1 放在左上角，置其余元素为 0. 用这样的矩阵对 S 进行扰动后，经 svd 分解后得到的旋转矩阵 $\mathbf{U}, \mathbf{V}$ 只会在前两行与前两列发生改变，所以这本质上仍然是一个 2 维的问题。

因此，不难给出所有的特征矩阵：

$$\Sigma_{ij} = \begin{array}{c} \\ \vdots \\ i \\ \vdots \\ j \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccc} \cdots & i & \cdots & j & \cdots \end{array} \\ \begin{bmatrix} \ddots & & \cdots & & \\ & 0 & \cdots & (-1)^{j-i} & \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ & (-1)^{i-j} & \cdots & 0 & \\ & & \cdots & & \cdots \end{bmatrix} \end{array}$$

每一个特征矩阵都表示在 $\mathbf{e}_i$ 与 $\mathbf{e}_j$ 所张成的子平面上的旋转。

4 实验

4.1 Proj. Newton 的运行结果

在 cow 模型上运行 Proj. Newton Method, 53 步后达到收敛。Newton Method 总用时 2.6s，平均每步迭代用时 0.05s

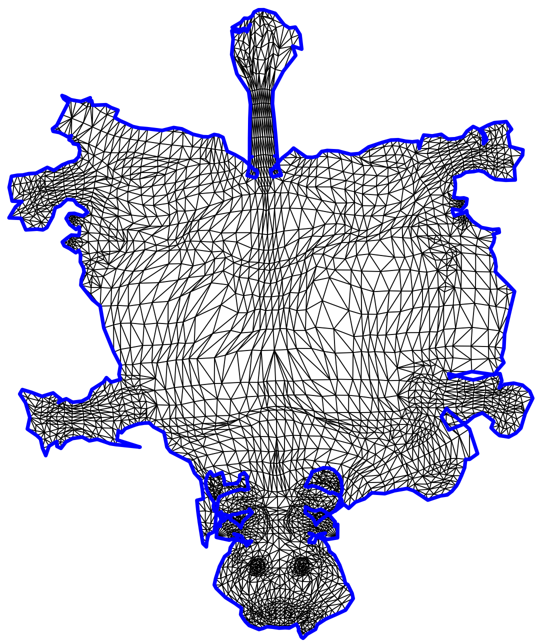
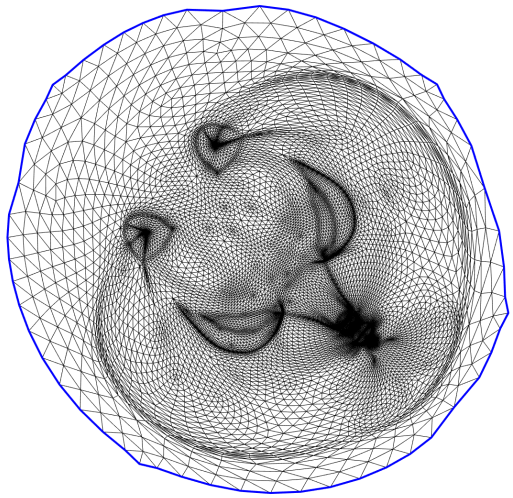


Figure 2: Proj. Newton Method 在 cow 上的运行结果

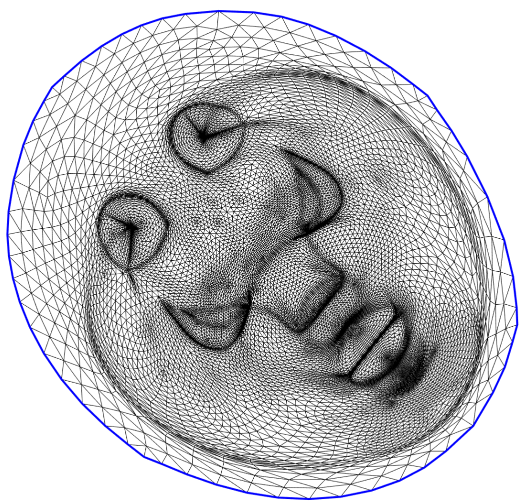
4.2 SLIM 与 Proj. Newton 的对比

我们在 camelhead 上对比 SLIM 与 Proj. Newton 方法，以 cotangent Laplacian 权重的 tutte 参数化为初始值，以梯度的无穷范数小于 10^{-4} 为收敛条件，Proj. Newton 方法在 24 步后就达到收敛，而 SLIM 方法需要 500 步左右才能收敛，符合原文附加材料中的实验结果。

由图 3，SLIM 在迭代的初期可以使能量迅速下降，但后期的收敛速度比较慢，而 Newton 法在初期能量的下降速度较 SLIM 缓慢，但最终收敛得更快。



(a) Newton, energy=63.5



(b) SLIM, energy=9.7

Figure 3: Newton 与 SLIM 在 5 步迭代后的结果对比

4.3 Hilbert Curve

我们在 Hilbert Curve 上进行迭代优化的参数化，参考 [2] 中的实验，初始使用 cotangent Laplacian weighted tutte 参数化，并进行 1 步 Newton 迭代，再进行 20 次 SLIM 迭代，最后经过 280 次 Proj. Newton 迭代达到收敛。

如图 5 所示的结果，SLIM 倾向于将 Hilbert 曲线从初始的圆盘状态（Tutte 参数化）拉平为一个长线条，从而使能量迅速下降，后续的 Newton 方法则倾向于将这个长线条卷曲起来，最终成为一个平面 Hilbert 曲线。

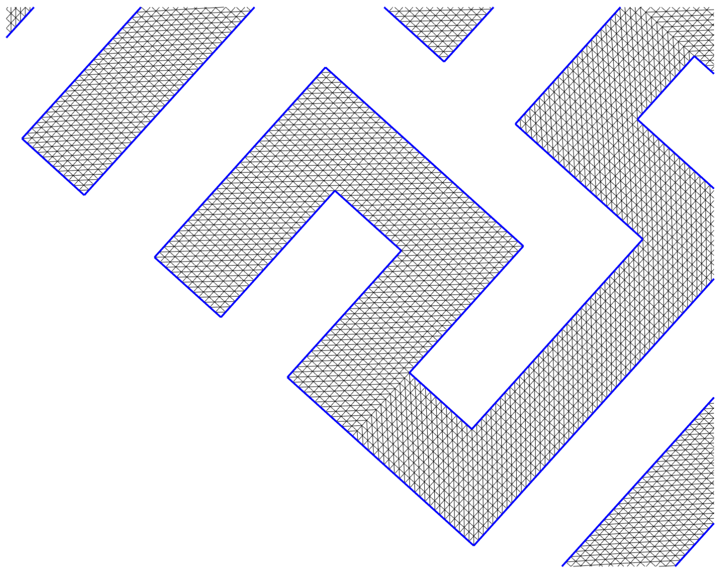


Figure 4: 图为 Hilbert Curve 参数化结果（局部），我们在迭代优化过程中验证了没有任何一个三角形发生翻转

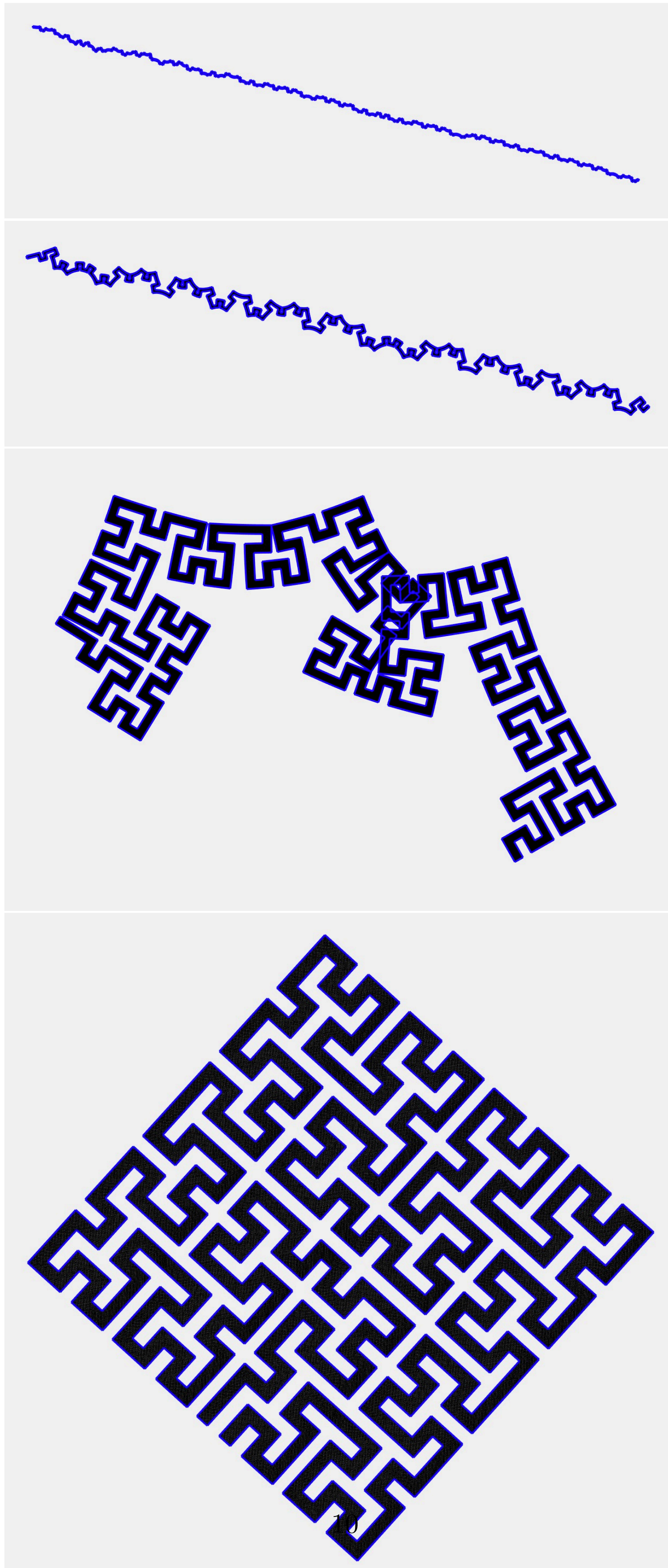


Figure 5: Hilbert Curve 的参数化过程，从上至下分别为进行 20 slim，20 slim + 几步 newton，20 slim + 120 newton，20 slim + 260 newton 的结果

References

[1] Breannan Smith, Fernando De Goes, and Theodore Kim. Analytic eigensystems for isotropic distortion energies. *ACM Trans. Graph.*, 38(1), feb 2019.

[2] Michael Rabinovich, Roi Poranne, Daniele Panozzo, and Olga Sorkine-Hornung. Scalable locally injective mappings. *ACM Trans. Graph.*, 36(2), apr 2017.

[3] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, 2e edition, 2006.