



LoopStructural 1.0: 时间 感知的地质建模

Surface 表面/界面

Foliation

Orientation 方位

| i

Support ???

stratigraphy



LoopStructural 1.0: 时间感知的地质建模

本文侧重软件架构和实施的介绍。

time-aware: 时间感知的

本文介绍**6种插值算法**，包括：

3个离散插值算法和3个多项式趋势插值算法，用于相同的建模设计。

LoopStructural 1.0可对不同的**地质对象**，如conformable foliations，断层面和不整合接触，混合使用不同的算法。

使用时间感知方法在建模中考虑地质特征，就是**最现代的特征最先建模**，用来约束更古代的地质特征整体的几何建模。



引言

不同地质特征之间的拓扑关系，诸如：horizons, faults interactions, intrusions and unconformities，对不同的模型组件使用多种隐式函数。

隐式建模有2种方法：

(1) **数据支持**的方法：在数据点处，计算基函数(Lajaunie et al., 1997)

(2) **离散插值**的方法，基函数在预定义的**support**上。

商业软件（如Petrel, Leapfrog等）使用的建模算法，一般仅提供一种插值算法，难以比较不同插值格式的效果；并且是“黑盒子”算法，修改算法参数的功能有限，无法了解算法是怎么实施的。

最近，开源的**GemPy库** (de la Varga et al., 2019)使用高性能计算库，实施**双co-Kriging隐式插值**算法，还有基于**机器学习**的隐式建模插值。

LoopStructural，基于Laurent et al. (2016) and Grose et al. (2017, 2018, 2019)的不断贡献，实施3D地质建模。核心库依赖于**SciPy, numpy, pandas**和一些科学计算的Python库；可视化模块使用**LavaVu** (Kaluza et al., 2020)，一个小型的OpenGL可视化软件包，允许在**Jupyter Notebook**环境中可视化模型。



材料与方法

2.1 隐式界面建模

隐式表面建模使用函数 $f(\mathbf{xyz})$ 表征地质特征的几何。使用2种方法：

- (1) 使用局参考水平面的距离作为标量场；
- (2) 势场方法。

隐函数使用基函数的加权形式表征：
$$f(\mathbf{xyz}) = \sum_{i=0}^N w_i \cdot \varphi_i(x, y, z)$$

有2种方法近似隐函数：

- (1) 使用离散公式的插值，其中 N 定义为某种网格；
- (2) 数据支持的基函数方法，其中 N 为数据点的数目。



2.1 隐式界面建模

在3D建模中考虑地质观测可分为2种：

(1) 描述地质特征方位(Orientation)的观测，如：on contact and off contact

(2) 描述地质特征位置的观测 (cumulative thickness for conformable stratigraphic horizons, or location of fault surface)

在地质图中，**location observations** may be the trace of a geological surface on the geological map, or a single point observation at an outcrop or from a borehole.

Orientation observations generally record a geometrical property of the surface – e.g. a vector that is tangential to the plane or the vector that is normal to the plane (black and dashed arrows in Fig. 1).

2.1.1 输入数据

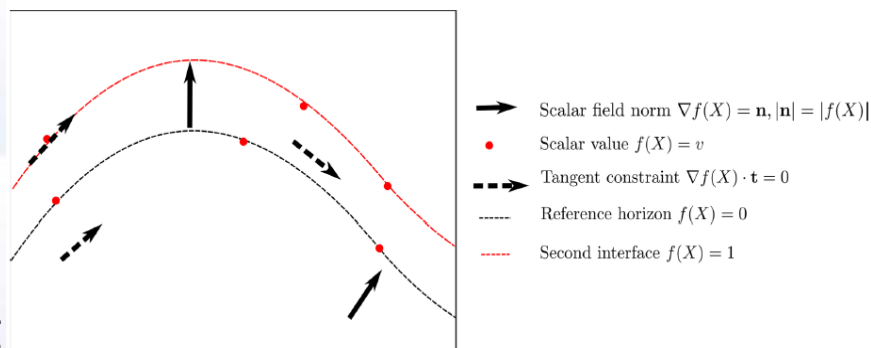


图1 显示不同类型插值约束的示意图，在2D上实施隐式插值格式。有2个界面：值为0的参考面，值为1的临近界面。这里展示了3类约束：(1) 标量场法向约束标量场的方位以及该位置处的隐函数的范数；(2) 标量场值约束控制标量场的值；(3) 切向约束仅约束隐函数的方位，不约束范数。修改自Hillier et al. (2014)



2.1.1 输入数据

当使用势场方法建模时，通过法向控制点的大小推测标量场。

使用符号距离方法时，观测值定义标量场，有效控制层厚度。

方位约束，要么控制方位的分量，即：指定函数的梯度应该与观测点以及隐函数的梯度法向正交。

在模型中的一个位置处，所有地质观测约束隐函数的一个分量：

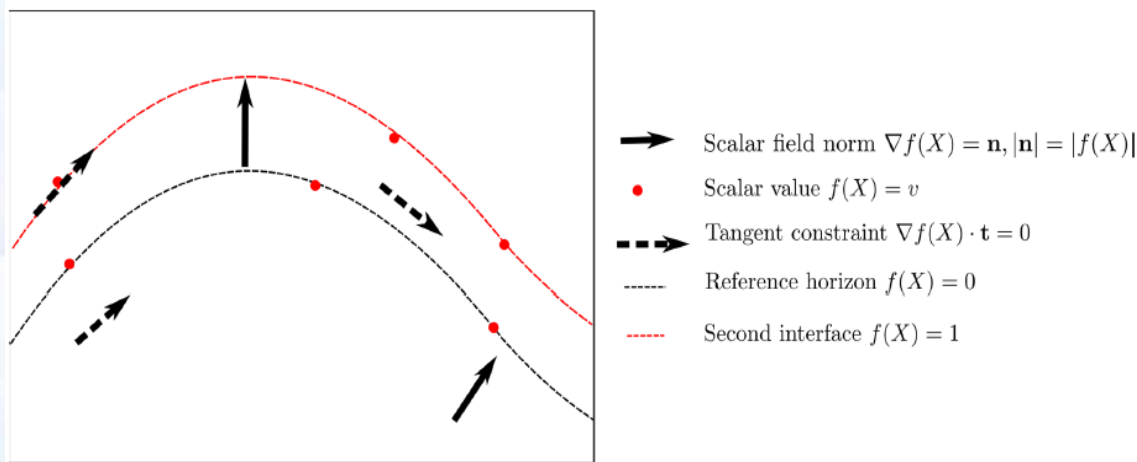


图1

(1) 地质特征位置的观测将约束标量场 $f(x,y,z)=v$ 的值。

(2) 接触方位观测可以：

- 1) 约束函数的偏微分 $\nabla f(x, y, z) = \mathbf{n}$
- 2) 约束与接触平行的一个向量 $\nabla f(x, y, z) \cdot \mathbf{t} = 0$



2.1 隐式界面建模

2.1.2 分段线性插值

由在3D四面体网格上分段线性函数定义体标量场。LoopStructural创建3D四面体网格是通过分割一个规则笛卡尔网格为四面体网格，其中一个立方体分为5个四面体（见附录A）。

2.1.3 有限差分插值

2.1.4 求解离散插值

2.1.5 数据支持的插值

使用相同位置处的基函数作为数据点做隐式面建模。

LoopStructural使用SurfE, C++程序，实施通用型径向基函数插值。SurfE有3种隐式表面重构：

- （1）使用径向基函数的符号距离插值；
- （2）使用双Co-Kriging的势场插值；
- （3）对各个面使用独立的标量场的符号距离插值。

LoopStructural与SurfE的接口允许用户使用所有的插值参数，这也包括了可使用更多复杂的求解器以及插值中添加光滑参数。



中國地質大學
China University of Geosciences

艰苦朴素 求真务实

温家宝

艰苦朴素
求真务实
温家宝

中国地质大学



2.2模拟地质特征

在地质模型中，岩石几何的软件相互作用，有**3种**方法：

- (1) 分层接触--沉积层之间的接触
- (2) 断层接触
- (3) 侵入接触

这些地质界面可以被变形构造影响，诸如 folds, faults and shear zones



2.2 模拟地质特征

2.2.1 沉积层接触(Stratigraphic contacts)

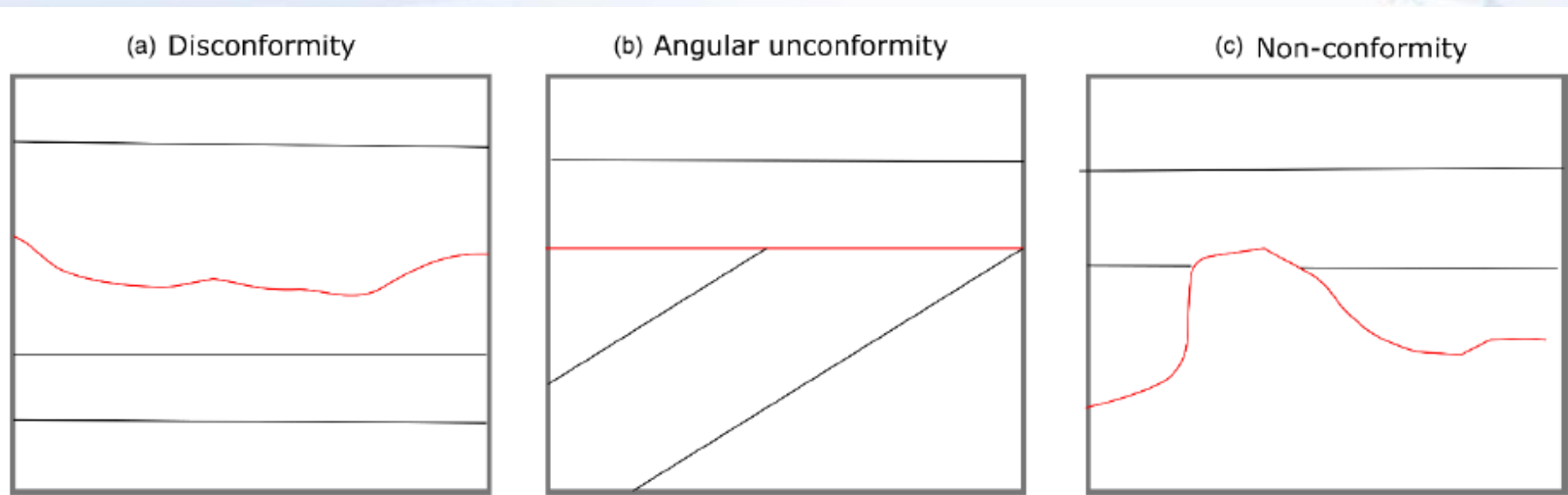


图2 不整合接触（红线）和地质界面（黑线）表征沉积历史的间断

一个不整合接触有不同可能的几何：(a) disconformity; (b) angular unconformity; (c) nonconformity



2.2 模拟地质特征

2.2.2 构造框架

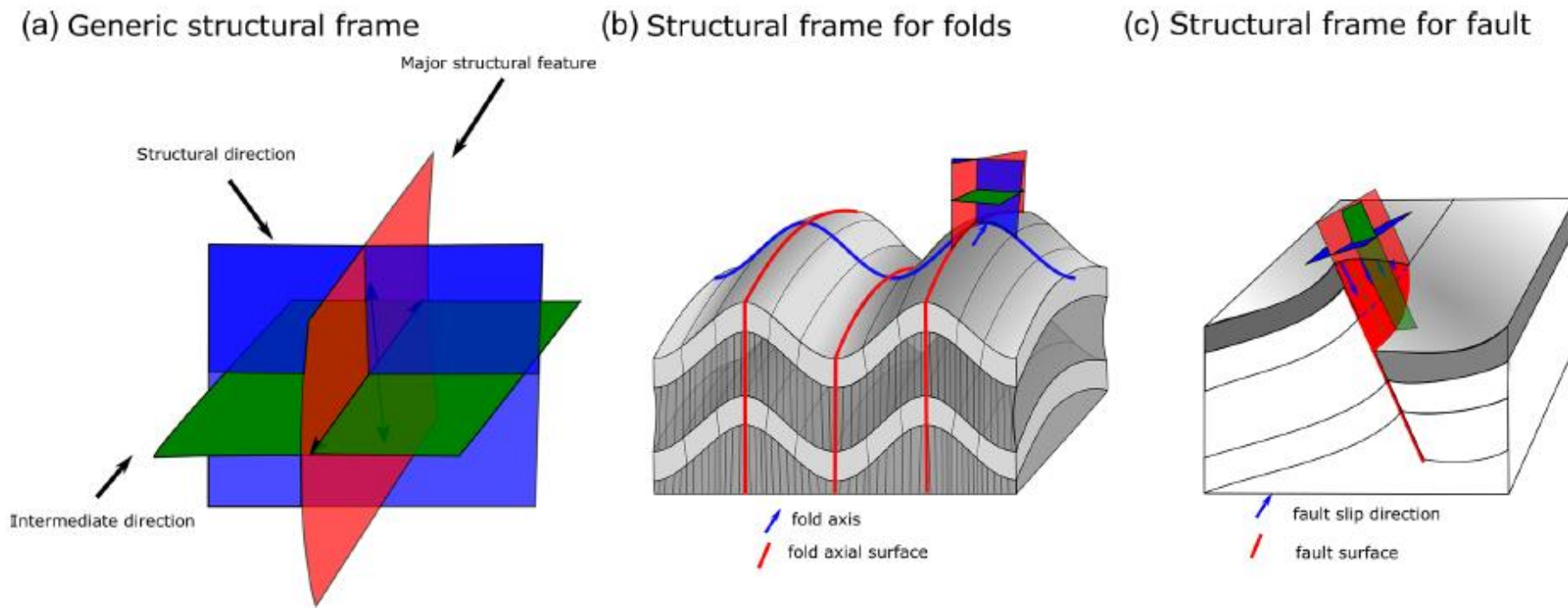


图3 (a) 构造框架显示3个坐标方向上的等值面；(b) 表征一个褶皱的构造框架；(c) 表征断层几何体的构造框架



2.2 模拟地质特征

2.2.3 断层

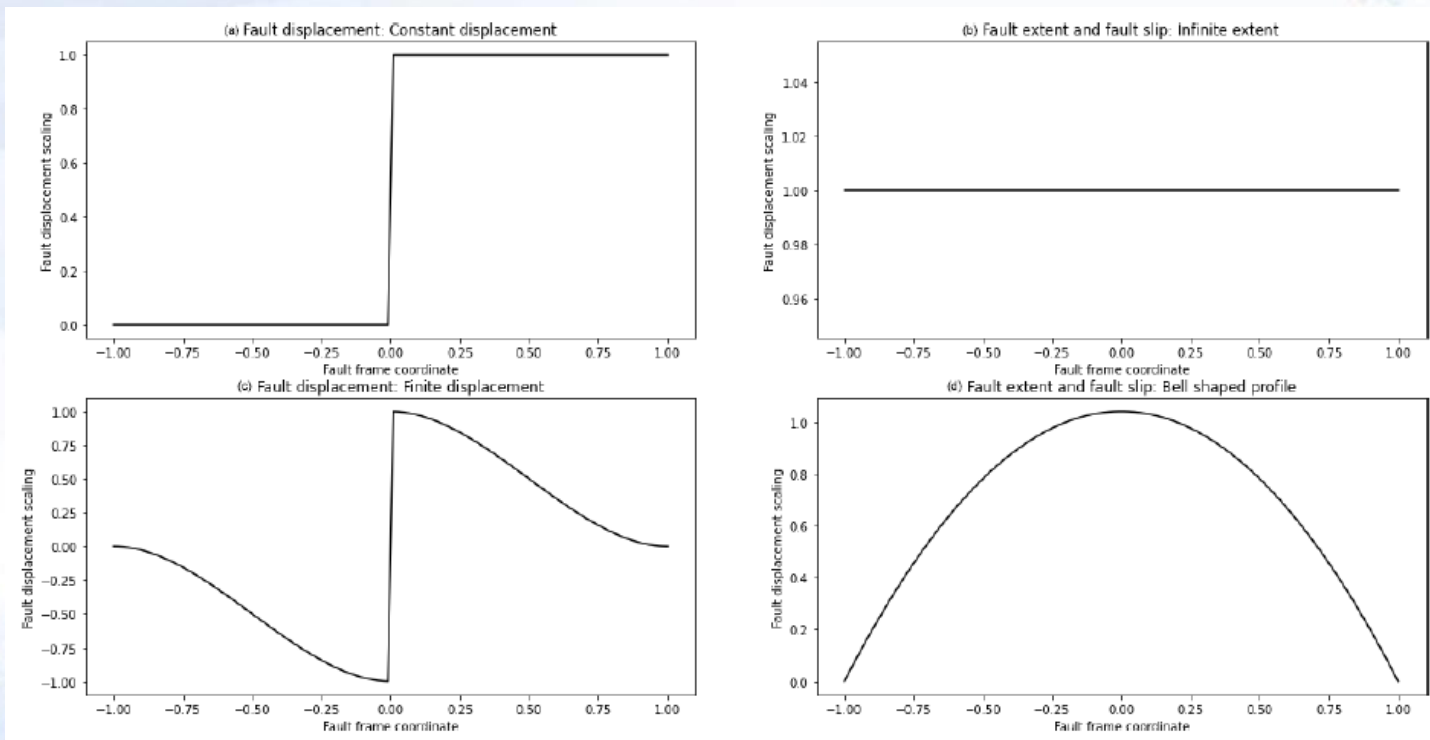


图4 断层位移剖面: (a) 恒定位移剖面; (b) 无穷范围的断层位移, 显示沿断层范围或在滑动方向上断层位移没有变化; (c) 固定范围的断层位移, 显示沿断层滑动方向的位移减小; (d) 固定范围钟型断层位移剖面, 保证沿断层范围方向或滑动方向上的断层未有



2.2 模拟地质特征

2.2.4 褶皱

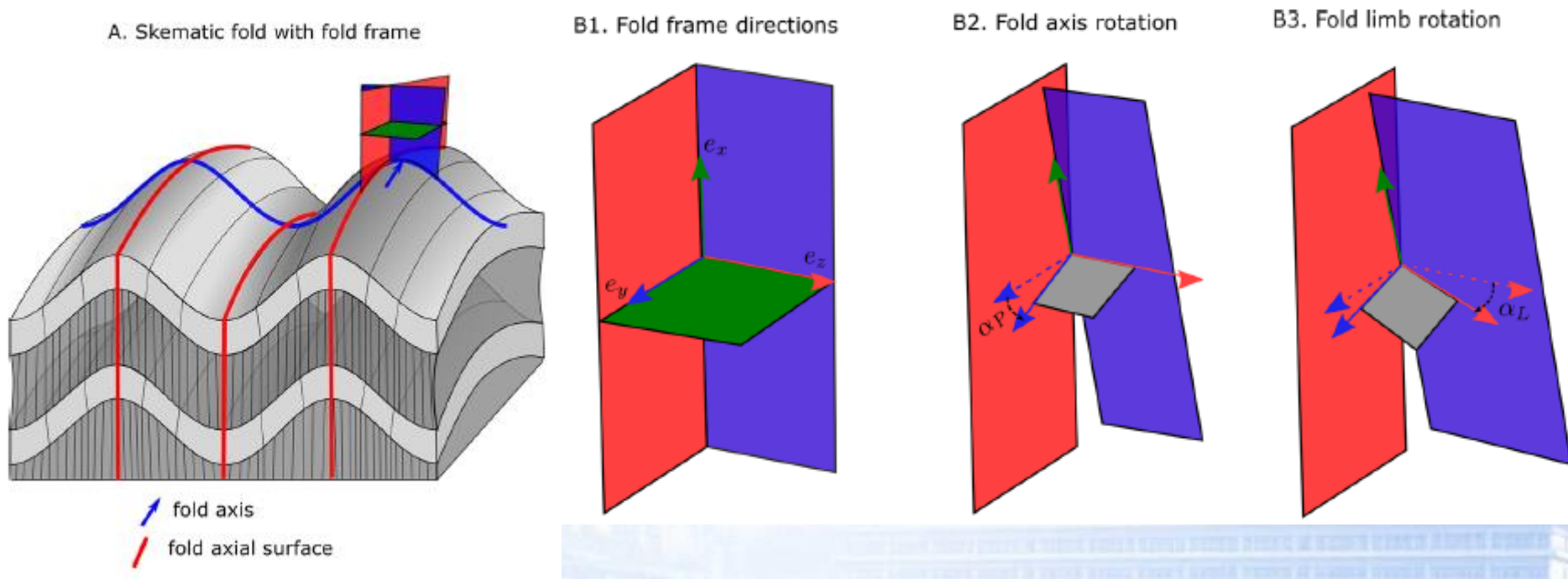


图5 褶皱: (a) 褶皱框架; (b1) 褶皱框架方向矢量; (b2) 褶皱轴向旋转角度定义的轴; (b3) 绕褶皱轴旋转的褶皱臂定义的褶皱foliation



3 LoopStructural的实施

3.1 Loop结构设计

Python3.6+, 使用Numpy数据结构和操作。

5个子模块:

- core
- interpolation
- datasets
- utils
- visualization



3 LoopStructural的实施

3.1 Loop结构设计

初始化实例，需要的参数：建模盒子边界的最大和最小范围，定义为2个独立向量，**rescale**系数：

$$\text{scale} = \max(x_{\max} - x_{\min}, y_{\max} - y_{\min}, z_{\max} - z_{\min})$$

通过使用**GeologicalModel**的实例，增加不同的地质对象。插值算法中可考虑4种不同类型的观测：

- (1) value
- (2) gradient
- (3) tangent
- (4) norm

与**GeologicalModel**相关的数据使用**set_data (data)**方法，其中**data**是**pandas**数据框架。向模型添加数据点时，将转换为模型的坐标系。



3 LoopStructural的实施

3.2 添加地质对象

- GeologicalFeatures表征所有的地质对象，包括：stratigraphy, faults, folding event, unconformities。一个GeologicalFeatures可以根据标量场的值and/or在某个位置上的标量场的梯度计算得到。
- GeologicalFeatures包含顺序的地质特征的集合，确定这些地质特征如何相互作用。
- GeologicalFeatures添加入GeologicalModel采用不同的方式，这要根据要模拟的地质特征类型。
- LoopStructural可以在相同模型中为不同的GeologicalFeatures指定不同的插值算法。通过为函数增加额外的关键词参数，指定插值算法和任何参数定义。



3 LoopStructural的实施

3.2 添加地质对象

表1 罗列了定义插值算法的可能参数。

Table 1. Interpolation keyword arguments. Default values are highlighted by bold text.

Keyword arguments	Description	Possible values
Interpolator_type	A choice for what interpolator to use	"PLP" , "FDI", "Surfe", "DFI"
solver	Which algorithm to solve the least squares problem (for PLI, FDI and DFI)	"cg" , "lu", "pyamg", "lsqr", "lsnr", "custom"
nelements	Number of elements in the discrete interpolation approach	100 : ∞ , 100 000
buffer	How much bigger to mesh around the model extents	0 : ∞ , 0.2
cpw	Weighting of value constraints in discrete least squares problem	0 : ∞ , 1
gpw	Weighting of gradient constraints in discrete least squares problem	0 : ∞ , 1
npw	Weighting of norm constraints in discrete least squares problem	0 : ∞ , 1
tpw	Weighting of tangent constraints in discrete least squares problem	0 : ∞ , 1
regularisation	Weighting of regularisation constraints in least squares problem	0 : ∞ , 1
data_region	Buffer around the observations to interpolate scalar field only on a subsection of the mesh	Boolean function, None



3 LoopStructural的实施

3.3 模型输出

在地质模型的坐标矩阵上计算 GeologicalModel。可以从一个 GeologicalModel调用如下函数：

(1) 为计算某位置处的 **岩性值 (lithology value)**，函数 `evaluate_model(xyz)` 返回包含 **地层编号** 的整型ID的numpy数组，定义为地层纵列。

(2) 为计算某位置处的 **GeologicalFeature 的值**，函数 `evalute_feature_value(feature_name, xyz)` 返回 **表征地质特征的标量场的值**。

(3) 为计算 **GeologicalFeature 的梯度**，可以调用 `evaluate_feature_gradient(feature_name, xyz)`。

三角化表面可以从 GeologicalFeature 提取，然后输出到常用的网格格式，如：**VTK (.vtk)** 或 **Wavefront (.obj)**。这些表面可以输入到外部软件，如 ParaView。



3 LoopStructural的实施

3.4 模型可视化

LoopStructural有3种可视化工具（LoopStructural.visualization模块）：

（1）LavaVuModelViewer: **LavaVu (Kaluza et al., 2020)**交互式可视化，使用LavaVu显示三角化表面，表征地质界面及描述隐函数的标量场。

（2）MapView: 使用matplotlib做2D可视化（断面，地图），从得到的地质模型创建一个地质图。使用**strike**和**dip**符号绘制接触的位置和方位。在地图面上计算标量场，绘制等值线或将地质模型绘制到地图上。

（3）FoldRotationAnglePlotter: 生成S制图和S-variogram制图的可视化模块，展示褶皱地质特征。使用matplotlib绘制。



4 示例

4.1 隱式表面建模

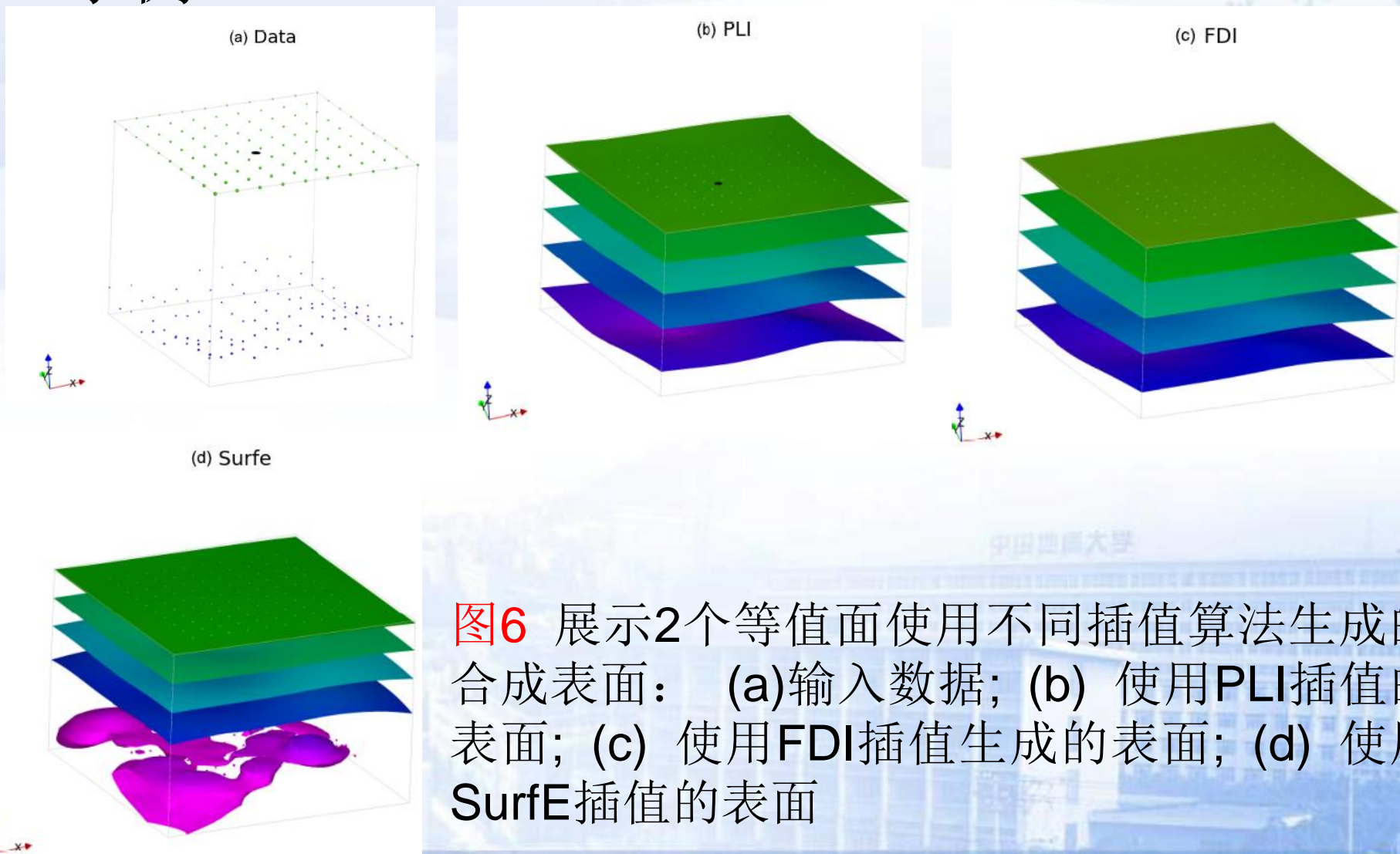


图6 展示2个等值面使用不同插值算法生成的合成表面： (a)输入数据；(b) 使用PLI插值的表面；(c) 使用FDI插值生成的表面；(d) 使用SurfE插值的表面



4示例

图6a显示数据点，图6b、图6c和图6d显示使用3种默认的插值算法生成的表面 (PLI, FDI, SurfE)。

当使用离散插值时，调整约束的权重通常对建模结果影响最大。如图7，.....

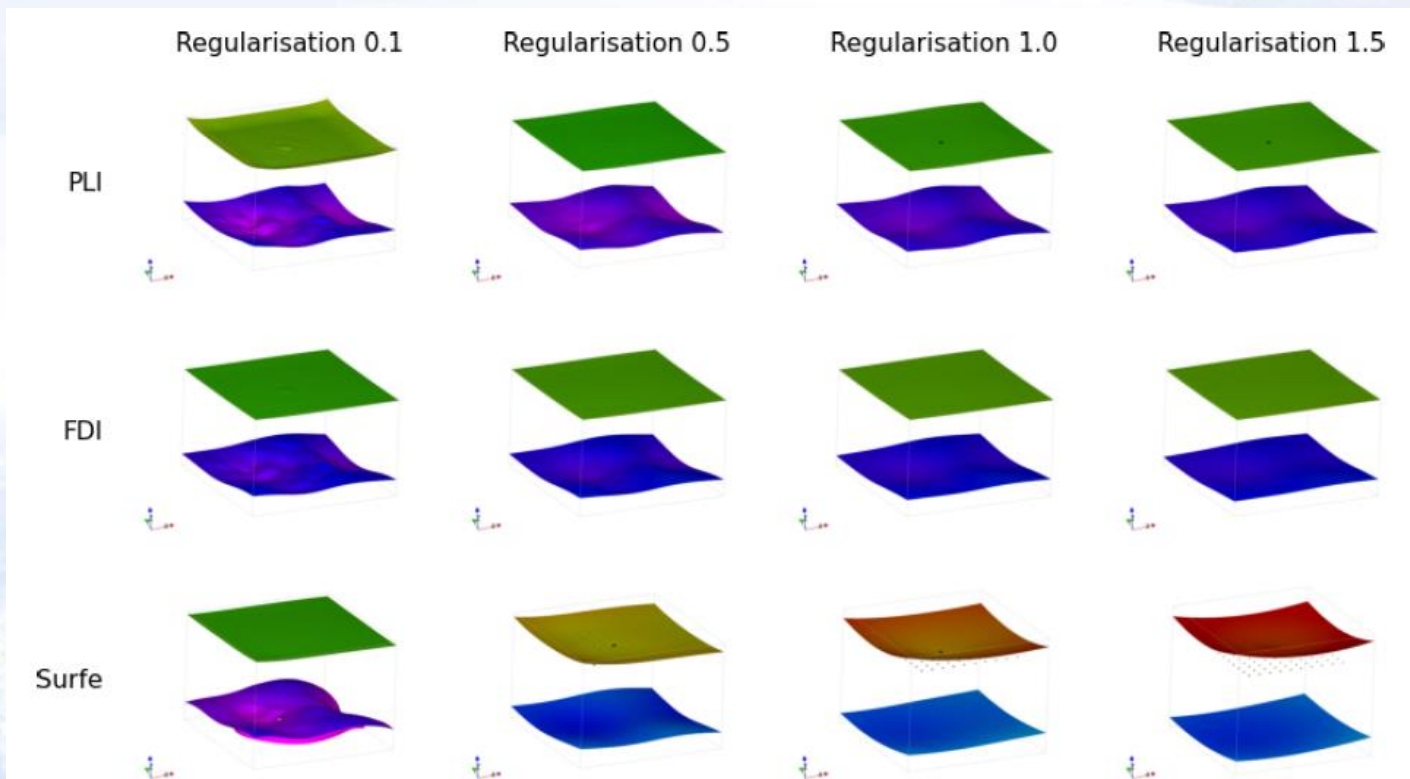


Figure 7. Implicit surfaces calculated for regularisation constraints (0.1,0.5,1,1.5) using piecewise linear interpolator (PLI), finite-difference interpolator (FDI) and radial basis function (SurfE).



4示例

4.2 模拟褶皱：类型3推理

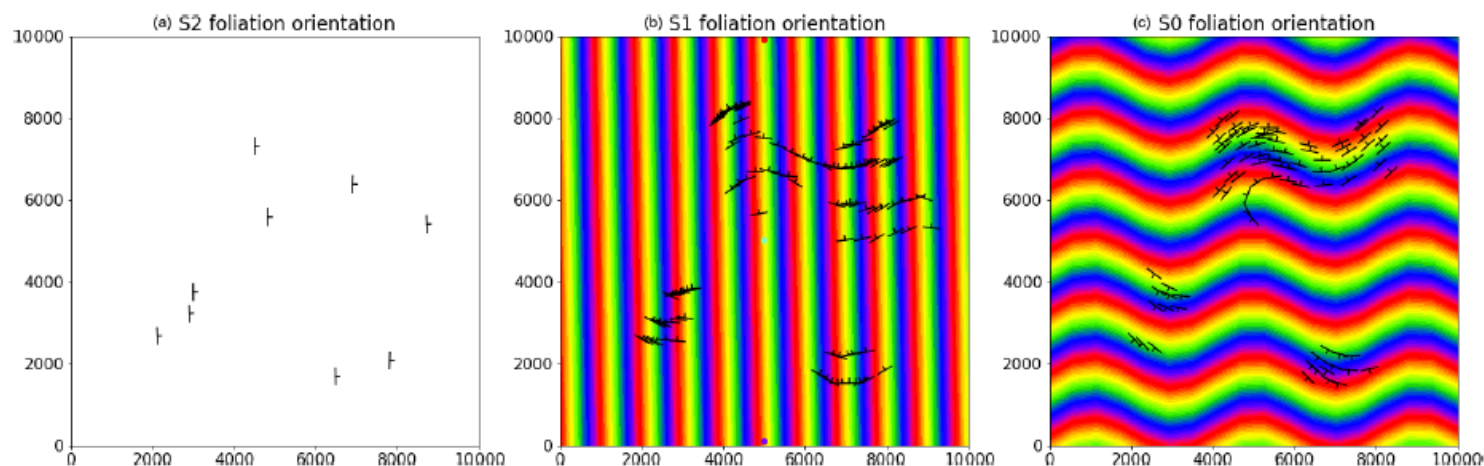


Figure 8. Structural data for refolded fold: (a) observations of S2, (b) observations of S1 showing interpolated scalar field of S2 and (c) observations of S0 showing interpolated scalar field of S1.

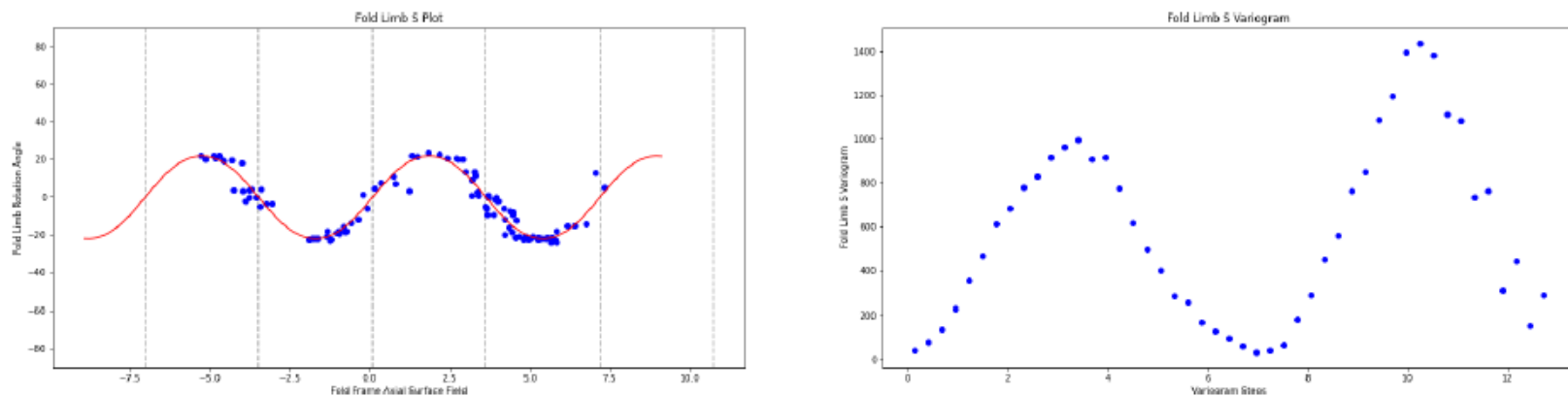


Figure 9. F2 S plot showing the fold rotation angle between observations of S1 and the fold frame S2.



4 示例

4.3 与map2loop的整合

使用map2loop (Jessell et al., 2021)作为前处理程序，从区域性地质调查图、国家地层数据和全球DEM，生成输入数据集。

map2loop创建一套augmented数据文件，用于在LoopStructural中构建地质模型。

类方法 `GeologicalModel.from_map2loop_directory(m2l_directory, **kwargs)`，从一个map2loop输出根目录，创建一个GeologicalModel实例。

下面通过2个实际案例展示map2loop与LoopStructural间的接口：

(1) Flinders Ranges in South Australia

(2) Hamersley region in Western Australia

第1个案例展示对大的区域性模型map2loop与LoopStructural间的接口。

第2个案例展示如何修改生成输入数据集的概念模型。



4示例

4.3 与map2loop的整合

第1个案例：模型面积85 km x 53 km。图12a展示层序单元；图12b展示地质图中的露头地质，没有任何地质单元的地图表示浅层Tertiary and Quaternary覆盖。map2loop从露头地质单元提取基部接触，计算分层厚度。

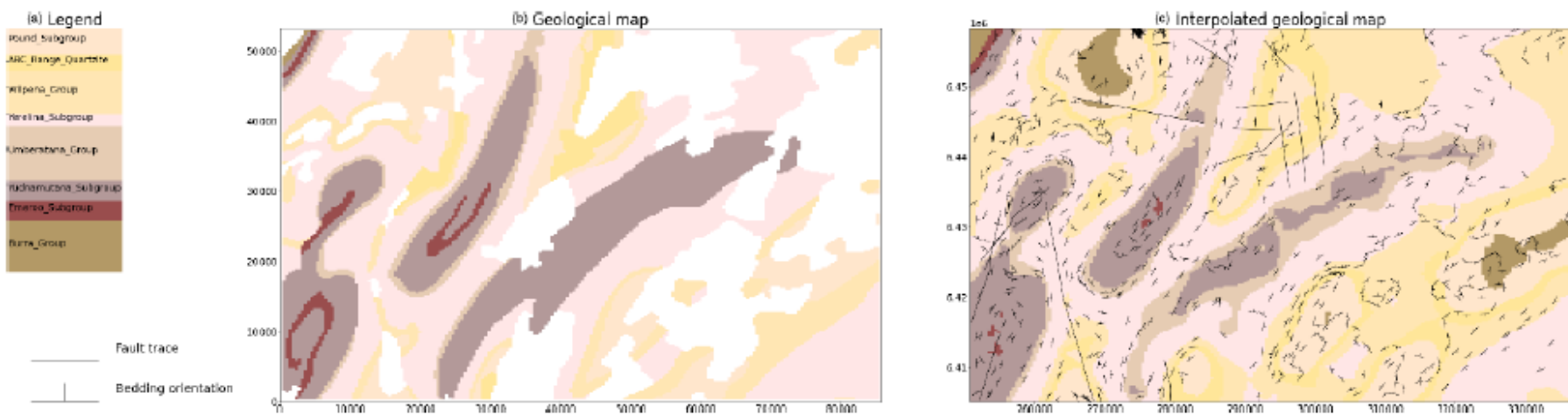


Figure 12. (a) Stratigraphic column for model area showing relative thickness. (b) Geological map showing bedding and faults. (c) Geological model shown on map surface.

建模工作流都封装在GeologicalModel.from_map2loop_directory(m2l_directory, **kwargs))类方法，意味着无需任何用户输入就能生成地质模型。



4 示例

4.3 与map2loop的整合

生成的地质模型如图13，表面表征一个地层组的基底，使用分层列着色（图12a）。模型中的断层使用50000个单元的笛卡尔网格插值，使用FDI插值算法，插值矩阵使用pyamg多重网格求解器。层序使用更细的网格（50万个单元），使用FDI插值及pyamg。i7处理器的笔记本（32Gb内存），使用map2loop花费近1min处理数据，渲染笛卡尔网格(200x2000x100)的表面花费3min。

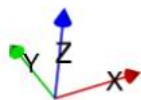
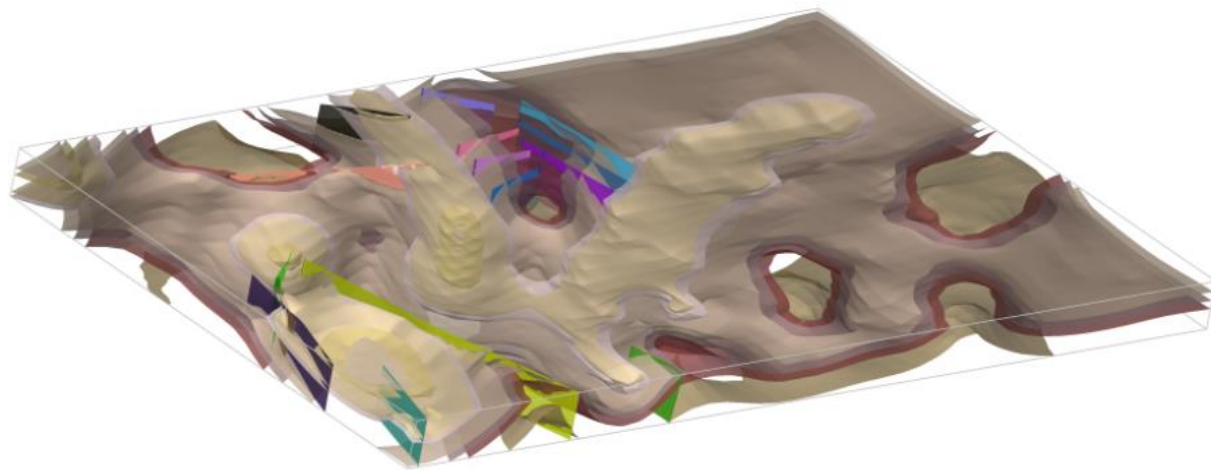


Figure 13. Geological model from South Australia using map2loop processed data stratigraphic surfaces using colours from Fig. 12a and fault surfaces.



4示例

4.3 与map2loop的整合

第2个案例：使用map2loop处理小范围的Turner Syncline in the Hamersley region in Western Australia，数据由Geological Survey of Western Australia提供。模型范围12km x 13km，包含3个断层。map2loop默认假设断层是垂直的、纯dip滑动、通过分析断层单元的地图模式计算hanging wall与位移（参考Jessell et al., 2021）。图14显示生成的地质模型，包括断层滑动向量和断层平面。



5 讨论

- (1) 时间感知的地质建模，采用与地质事件发生的相反顺序添加地质特征，这样能捕捉复杂构造几何，这种方法用于模拟refolded褶皱（图11）。
- (2) LoopStructural提供了灵活的开源实施隐式地质建模算法 workflows。这意味着可以开发新的隐式建模算法与工具，比如从外部网格生成代码导入LoopStructural，使用一个二次开发的类重写四面体网格类。
- (3) 反问题，模拟地质建模中的不确定性。
- (4) map2loop从开放的地质调查数据（层序数据集、DTM、地质shapefile、构造线和构造观测）生成增强数据集。从数据处理到模型渲染需要约10min。
- (5) 断层位移剖面的优化。断层滑移向量和断层倾角由概念模型或观测数据定义，但是指定约束断层滑移的具体的概率函数是有挑战的，与具体的地质问题有关，即当观测数据充足时，可使用Godefroy et al. (2018b)???的方法并从插值中分离部分数据；但是当处理典型的区域性比例尺地图层时，大部分观测仅在地表附近，约束3D地质几何是有限的。



附录A 四面体网格

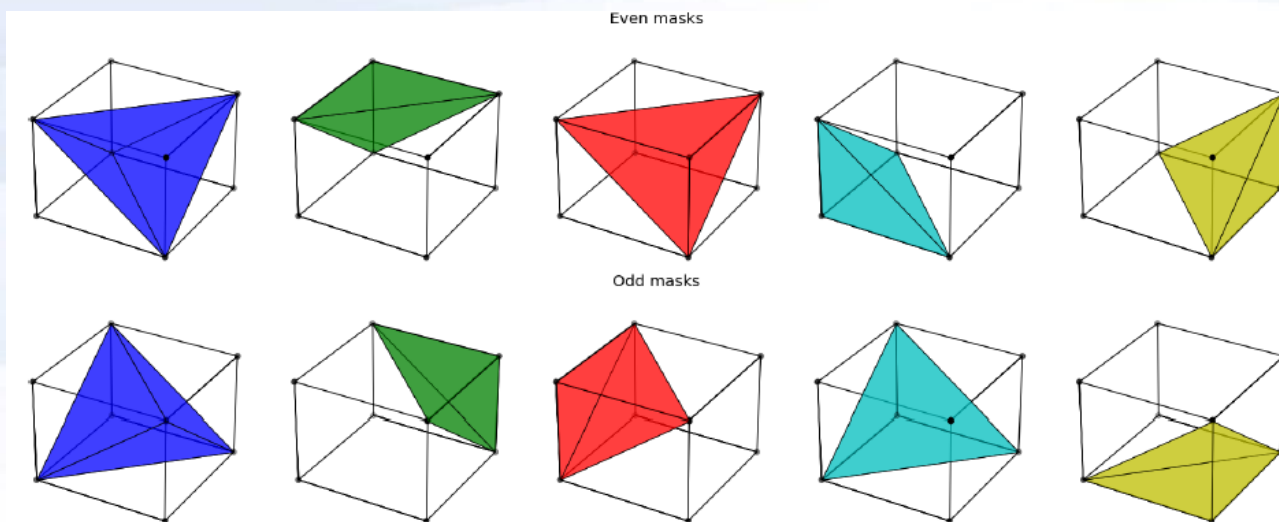


Figure A1. Odd and even masks for extracting tetrahedrons from a Cartesian grid.

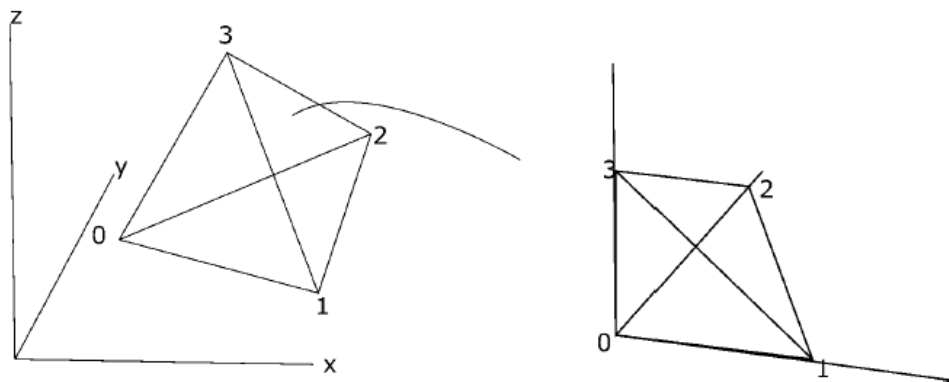


Figure A2. Schematic diagram showing transformation from tetrahedron in Cartesian space to reference tetrahedron in natural coordinates. This transformation allows for the shape functions and derivatives to be simplified.



附录B 立方体单元内的trilinear插值

隐式函数，可使用形函数($N_{0,...,7}$)相对单元的8个节点描述：

$$N_0 = \frac{1}{8} (1 - \xi) (1 - \eta) (1 - \zeta), \quad (\text{B1})$$

$$N_1 = \frac{1}{8} (1 + \xi) (1 - \eta) (1 - \zeta), \quad (\text{B2})$$

$$N_2 = \frac{1}{8} (1 + \xi) (1 + \eta) (1 - \zeta), \quad (\text{B3})$$

$$N_3 = \frac{1}{8} (1 + \xi) (1 - \eta) (1 + \zeta), \quad (\text{B4})$$

$$N_4 = \frac{1}{8} (1 - \xi) (1 - \eta) (1 + \zeta), \quad (\text{B5})$$

$$N_5 = \frac{1}{8} (1 + \xi) (1 - \eta) (1 - \zeta), \quad (\text{B6})$$

$$N_6 = \frac{1}{8} (1 + \xi) (1 + \eta) (1 + \zeta), \quad (\text{B7})$$

$$N_7 = \frac{1}{8} (1 - \xi) (1 + \eta) (1 + \zeta). \quad (\text{B8})$$