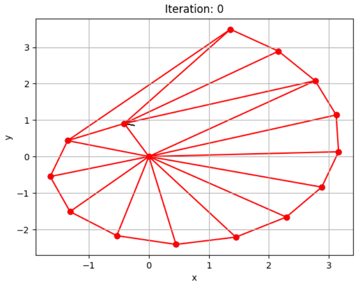
Research Update

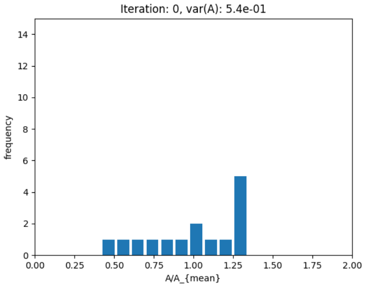
Rajarshi Dasgupta

PhD student
STARS

June 21, 2024

Reducing the variation in the area distribution





Reducing the variation in the area distribution

We aim to minimise the variation in the area distribution without changing the coordinates of the boundary vertices. We define the loss function and apply the gradient descent approach.
The loss function takes as input the x and y coordinates of the triangles and gives us the variation in the distribution of the area of the triangular elements.

```
def loss(x_tri, y_tri):
    x02 = x_tri[0::3] - x_tri[2::3]
    x12 = x_tri[1::3] - x_tri[2::3]

    y02 = y_tri[0::3] - y_tri[2::3]
    y12 = y_tri[1::3] - y_tri[2::3]

    area = x02*y12 - x12*y02
    mean_area = tf.reduce_mean(area)
    return tf.reduce_mean(tf.square( area - mean_area ))
```

Reducing the variation in the area distribution

While applying the gradients we make sure that elements corresponding to the same vertex do not have different values. This is ensured by operating a sparse tensor on the gradients to modify them for this requirement.

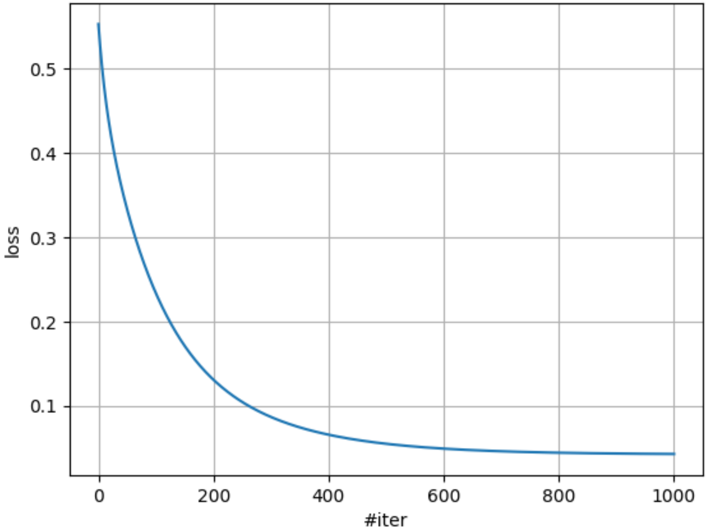
```
for iter_id in range(iter_max):
    with tf.GradientTape() as tape:
        var_area = loss(x_tri_tf, y_tri_tf)
    gradients = tape.gradient(var_area, [x_tri_tf, y_tri_tf])

    loss_np = var_area.numpy()
    losses.append(loss_np)

    # Operate on gradients to equalise entries for same vert
    grad_x_apply = tf.sparse.sparse_dense_matmul(eq_op_sp_tf, gradients[0])
    grad_y_apply = tf.sparse.sparse_dense_matmul(eq_op_sp_tf, gradients[1])

    x_tri_tf.assign_sub(learning_rate * grad_x_apply)
    y_tri_tf.assign_sub(learning_rate * grad_y_apply)
```

Reducing the variation in the area distribution

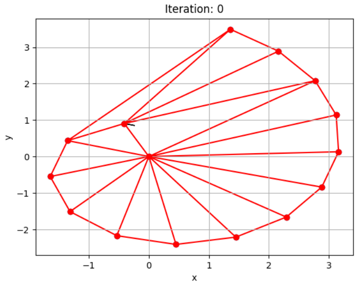
The sparse tensor is defined before the gradient descent process begins.

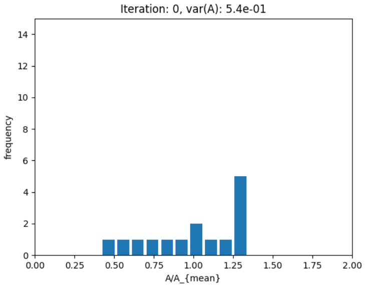
```
indices = []
values = []
for vert in interior_vertices:
    instances = np.where(ind_tri_flat == vert)
    val = 1.0/instances[0].shape[0]
    for ind1 in instances[0]:
        for ind2 in instances[0]:
            values.append(val)
            indices.append([ind1,ind2])
eq_op_sp_tf = tf.sparse.SparseTensor(indices, values, [n_tri3,n_tri3])
```

Reducing the variation in the area distribution

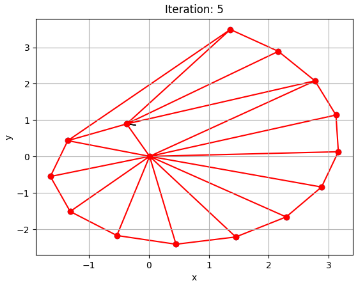


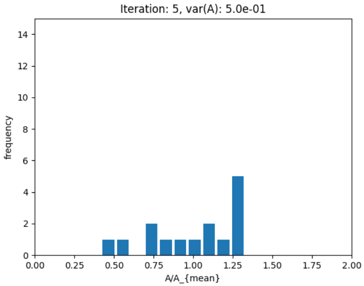
Reducing the variation in the area distribution



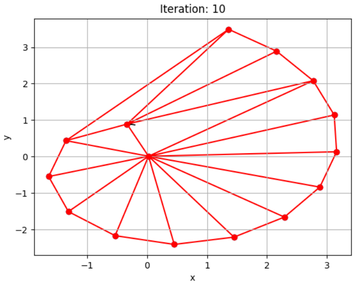


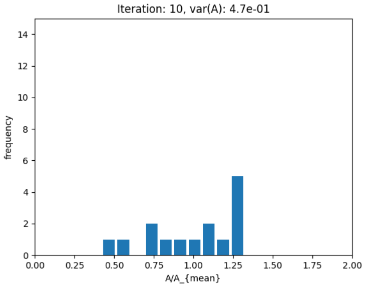
Reducing the variation in the area distribution



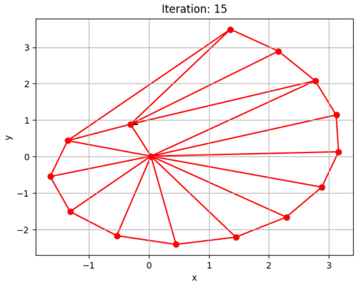


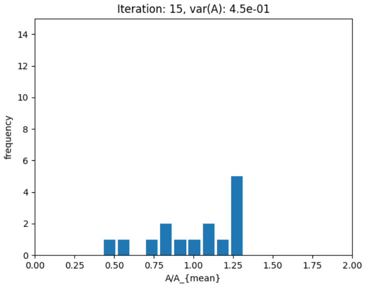
Reducing the variation in the area distribution



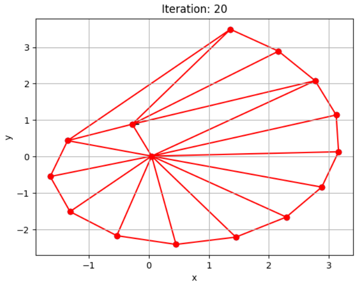


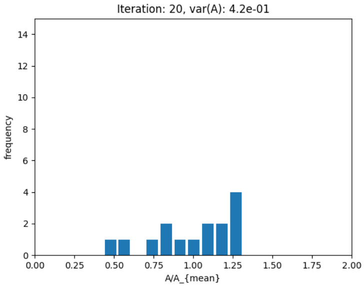
Reducing the variation in the area distribution



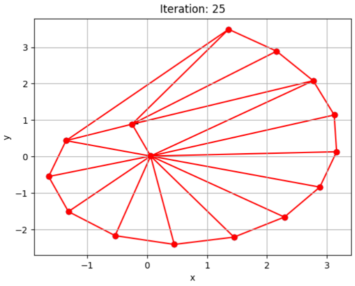


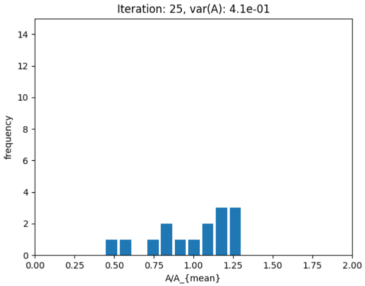
Reducing the variation in the area distribution



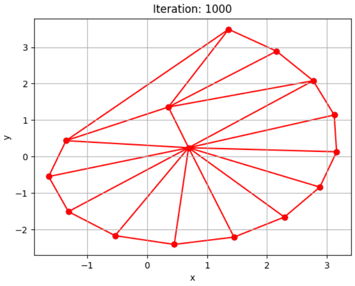


Reducing the variation in the area distribution

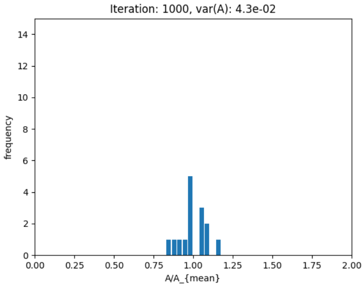




Reducing the variation in the area distribution



Iteration: 1000
y
x
−1
0
1
2
3
−2



Iteration: 1000, var(A): 4.3e-02
frequency
A/A_{mean}
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
0
2
4
6
8
10
12
14