JAX FDM
A differentiable framework for constrained form-finding

Rafael Pastrana, Sigrid Adriaenssens

https://github.com/arpastrana/jax_fdm
Form-finding is shape optimization

A form-finding method computes a shape in static equilibrium

Tension-only

Compression-only

Photo credit: Adobe stock
Form-finding is shape optimization

A form-finding method computes a shape in static equilibrium
Form-finding is shape optimization

A form-finding method computes a shape in static equilibrium

\[ \sum F_i = 0 \]
The force density method (FDM)

A numerical form-finding method for vaults and cable-nets

The force density method (FDM)

How does it work? Forward
The force density method (FDM)

How does it work? Define the force density of the bars

\[ q_{i,j} = \frac{\text{force}_{i,j}}{\text{length}_{i,j}} \]
The force density method (FDM)

How does it work? Forward

The force density method (FDM)

How does it work? Get the XYZ coordinates of the nodes

\[ u_i = [0.53, 0.49, 2.83] \]

The force density method (FDM)

A plethora of shapes in static equilibrium for different force densities

The challenges of form-finding

Viable shapes in static equilibrium meet additional constraints: fabrication, structural, aesthetic, and environmental.

No two structures are (or want to be) alike.
The challenges of form-finding

Shapes in equilibrium conditioned on fabrication constraints
The challenges of form-finding

Capturing architectural design intent

Constrained form-finding is the challenge

We actually want to solve an inverse problem

“What is the set of force densities $q$ that are best conducive to it?”

“A shape in static equilibrium which approximates this other shape”

Constrained form-finding by hand?

Constrained form-finding by hand is laborious and error prone.

JAX FDM

A differentiable, hardware-accelerated framework for constrained form-finding in structural design.

Crafted with care in the Form-Finding Lab at Princeton University 💖

JAX FDM enables the solution of inverse form-finding problems for discrete force networks using the force density method (FDM) and gradient-based optimization. It streamlines the integration of form-finding simulations into deep learning models for machine learning research.

Key features

- Differentiable formulation
- Hardware-accelerated
- Constrained form-finding
- Force density method
- Gradient-based optimization

URL: https://github.com/arpastrana/jax_fdm
JAX FDM A solution

A differentiable tool for constrained form-finding. Powered by JAX

JAX FDM A solution

A differentiable tool for constrained form-finding. Powered by JAX

We apply backpropagation for the inverse design of 3D structures

\[ q^t = q^{t-1} - \lambda \nabla_q L(u) \]

\[ F_q(T, x) \]

\[ L(u) = \|u - \bar{u}\|^2 \]

Constrained form-finding Backpropagation

Inverse design with backpropagation is faster and more stable

from jax import grad
import jax.numpy as jnp
from jax_fdm.datastructures import FDNetwork
from jax_fdm.equilibrium import EquilibriumModel

# create the FDM model
pattern, supports, loads = FDNetwork.from_json('arch.json')
model = EquilibriumModel(pattern, supports, loads)

# define the loss function
def loss(q, target_length=1.5):
    eq_state = model(q)
    return jnp.mean((eq_state.edge_lengths - target_length)**2)

# vanilla gradient descent
q = jnp.ones(10) * -1.0  # initial guess
lr = 0.1  # step size
for i in range(1000):
    loss_value = loss(q)
    q = q - lr * grad(loss)(q)
from jax import grad
import jax.numpy as jnp
from jax_fdm.datastructures import FDNetwork
from jax_fdm.equilibrium import EquilibriumModel

# create the FDM model
pattern, supports, loads = FDNetwork.from_json('arch.json')
model = EquilibriumModel(pattern, supports, loads)

# define the loss function
def loss(q, target_length=1.5):
    eq_state = model(q)
    return jnp.mean((eq_state.edge_lengths - target_length) ** 2)

# vanilla gradient descent
q = jnp.ones(10) * -1.0  # initial guess
lr = 0.1  # step size
for i in range(1000):
    loss_value = loss(q)
    q = q - lr * grad(loss)(q)
Constrained form-finding of tall buildings

Leveraging static equilibrium to design new vertical structures

Constrained form-finding to reduce waste

Vaults that require minimal scaffolding during construction

Mexican architect

Mexican builder
Constrained form-finding to reduce waste

Vaults that require minimal scaffolding during construction

JAX FDM What's next?

Combining machine learning and differentiable form-finding

Learning to solve the inverse problem
JAX FDM as a layer in a neural network. 10x speed-up?

Learning continuous representations of graphs
Can we optimize patterns and force densities jointly?
Constrained form-finding is the challenge

We actually want to solve an inverse problem

"What is the set of force densities $q$ that are best conducive to it?"

"A shape in static equilibrium which approximates this other shape"

Next steps Learning to solve the inverse problem

Differentiable form-finding as a layer in a neural network
Next steps Learning to solve the inverse problem

Differentiable form-finding as a layer in a neural network

10x speed-up?
Next steps Generative model of patterns

Next steps Generative model of patterns

Learning continuous representations of graphs
Cast as a NLP task?

Next steps: Generative model of patterns

Learning continuous representations of dis...
Next steps **Generative model of patterns**

Learning continuous representations of graphs
Cast as a **NLP** task?

Next steps Generative model of patterns

Learning continuous representations of graphs
Cast as a NLP task?

JAX FDM
A differentiable framework for constrained form-finding

Rafael Pastrana, Sigrid Adriaenssens

https://github.com/arpastrana/jax_fdm