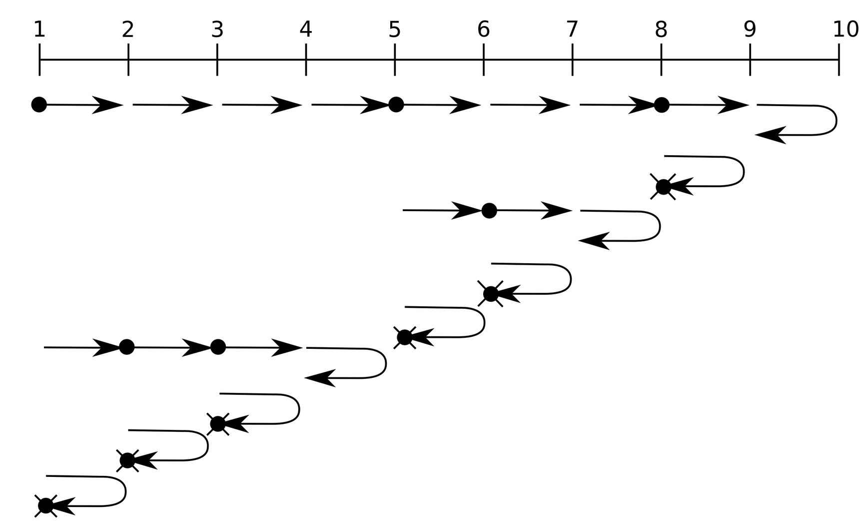
DOLFIN-ADJOINT
AUTOMATIC ADJOINT MODELS FOR FENICS

P. E. Farrell, S. W. Funke, D. A. Ham, M. E. Rognes
Simon@simula.no

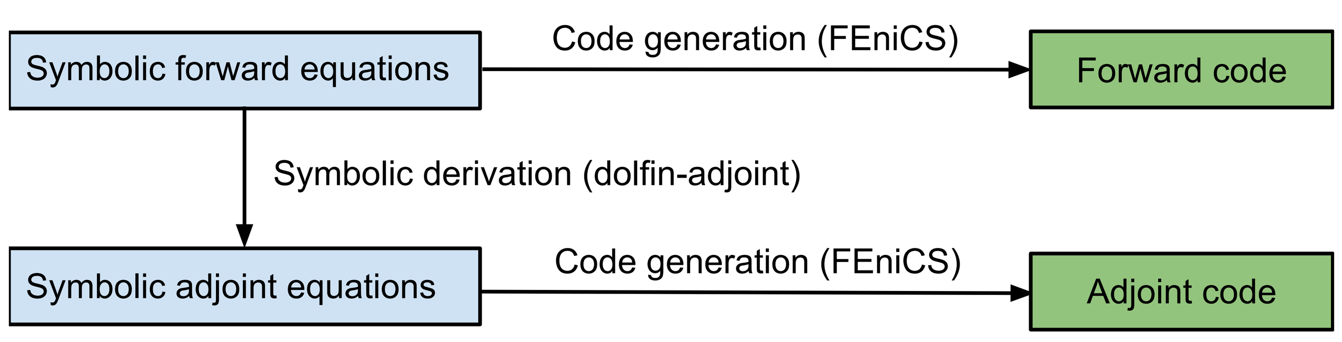
The dolfin-adjoint project automatically derives and solves adjoint and tangent linear equations from high-level mathematical specifications of finite element discretizations of partial differential equations.

How it works

Adjoint and tangent linear models form the basis of many numerical techniques, including sensitivity analysis, optimization and stability analysis. The implementation of adjoint models for nonlinear or time-dependent models are notoriously challenging: the manual approach is time-consuming and traditional automatic differentiation tools lack robustness and performance.
dolfin-adjoint solves this problem by automatically analyzing the high-level mathematical structure inherent in finite element methods. It raises the traditional abstraction of algorithmic differentiation from the level of individual floating point operations to that of whole systems of differential equations. This approach delivers a number of advantages over the previous state-of-the-art: robust hands-off automation of adjoint model derivation, optimal computational efficiency, and native parallel support.



△ **Figure:** dolfin-adjoint employs a binomial checkpointing strategy to trade off computing and storage requirements.



◁ **Figure**: dolfin-adjoint's implementation is based on the finite-element framework FEniCS. By adding a few lines of code to an existing FEniCS model, dolfin-adjoint computes tangent linear and adjoint solutions, gradients and Hessian actions of arbitrary user-specified functionals.

Example: Sensitivity analysis

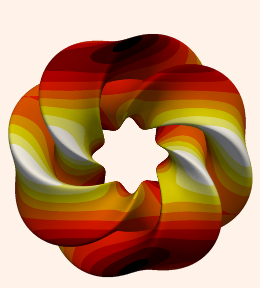
Consider the time dependent heat equation

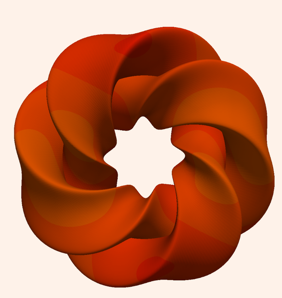
$$\begin{aligned} \frac{\partial u}{\partial t} - \nu \nabla^2 u = 0 \quad & \text{in} \quad \Omega \times (0, T), \\ u = g \quad & \text{for} \ \Omega \times \{0\}. \end{aligned}$$

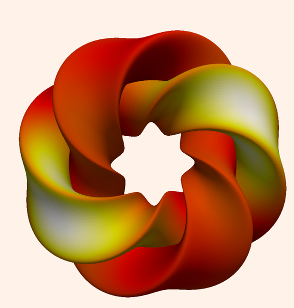
Here Ω is the Gray's Klein bottle, a closed 2D manifold embedded in 3D, T is the final time, u is the unknown temperature, ν is the thermal diffusivity, and g is the initial temperature.
The goal is to compute the sensitivity of the norm of temperature at the final time

$$J(u) = \int_\Omega u(t = T)^2 \,\mathrm{d}x$$

with respect to the initial temperature, that is $\mathrm{d}J/\mathrm{d}g$.


Initial temperature


Final temperature


Sensitivity

```
from dolfin import *
from dolfin_adjoint import *

# Solve the forward system
F = u*v*dx - u_old*v*dx +
    dt*nu*inner(grad(v),grad(u))*dx
while t <= T:
    t += dt
    solve(F == 0, u)

# Apply dolfin-adjoint
m = Control(g)
J = u**2*dx*dt[T]
dJdm = compute_gradient(J, m)
H = hessian(J, m)
```

△
Code: Implementation excerpt (the complete code has 37 lines)

Example: PDE-constrained optimization

This topology optimization example minimizes the compliance

$$\int_\Omega fT \,\mathrm{d}x + \alpha \int_\Omega \nabla a \cdot \nabla a \,\mathrm{d}x,$$

subject to the Poisson equation with mixed Dirichlet–Neumann conditions

$$\begin{aligned} -\mathrm{div}(k(a)\nabla T) = f \qquad & \text{in } \Omega, \\ T = 0 \qquad & \text{on } \partial\Omega_D, \\ k(a)\nabla T = 0 \qquad & \text{on } \partial\Omega_N, \end{aligned}$$

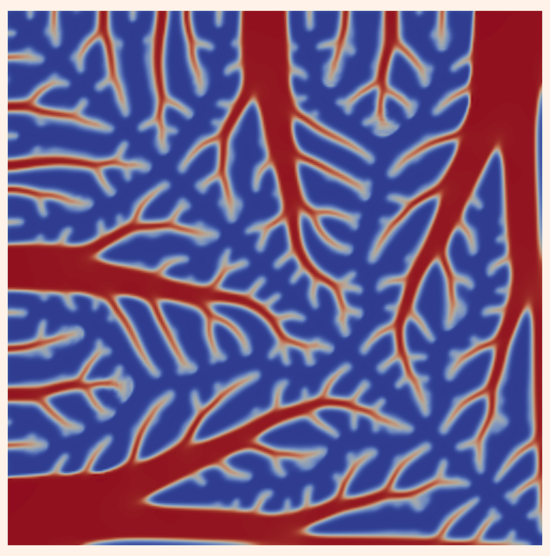
and additional control constraints

$$\int_\Omega a \,\mathrm{d}x \le V \text{ and } 0 \le a(x) \le 1 \qquad \forall x \in \Omega.$$

Here Ω is the unit square, T is the temperature, a is the control ($a(x) = 1$ means material, $a(x) = 0$ means no material), f is a source term, $k(a)$ is the Solid Isotropic Material with Penalisation parameterization, α is a regularization term, and V is the volume bound on the control. Physically, the problem is to find the material distribution a that minimizes the integral of the temperature for a limited amount of conducting material.

```
from dolfin import *
from dolfin_adjoint import *
# ...
J = f*T*dx + alpha*inner(grad(a),grad(a))*dx
m = Control(a)
rf = ReducedFunctional(J, m)
minimize(rf, method="SLSQP", bounds=...)
```

△
Code: Implementation excerpt (the full code uses the IPOPT optimization package and has 56 lines)



◁ **Figure**: Optimal material distribution a for a unit square domain and $f = 10^{-2}$

This poster Website References

Center for Biomedical Computing EPSRC Engineering and Physical Sciences Research Council NERC SCIENCE OF THE ENVIRONMENT