```
using Gridap
# Manufactured solutions
u(x) = x[1]^2 + x[2]
f(x) = -Δ(u)(x); g(x) = u(x)
# FE mesh (aka discrete model)
pmin = Point(0,0,0); pmax = Point(1,1,1)
cells=(8,8,8); order = 1
model = CartesianDiscreteModel(pmin, pmax, cells)
# FE Spaces
V0 = TestFESpace(model=model, reffe=:Lagrangian,
  valuetype=Float64, order=order,
  conformity=:H1, dirichlet_tags="boundary")
Ug = TrialFESpace(V0, g)
# Weak form
a(u,v) = ∇(u)⋅∇(v); l(v) = v*f
trian_Ω = Triangulation(model)
quad_Ω = CellQuadrature(trian_Ω, 2*order)
t_Ω = AffineFETerm(a,l,trian_Ω,quad_Ω)
# FE Problem and solution
op = AffineFEOperator(Ug,V0,t_Ω)
uh = solve(op)
# Output for visualization
writevtk(trian_Ω,"results",
  cellfields=["uh"=>uh,"grad_uh"=>∇(uh)])
```