Nonlinear elasticity and Newton's method in infinite dimensions

Songlan Nie*
Francis R. A. Aznaran†, Charles Parker‡
Mathematical Institute, University of Oxford

December 20, 2023

Abstract

In this report, we focused on elasticity models and their numerical solutions. We introduced the linear elasticity model and the St Venant–Kirchhoff (SVK) hyperelasticity model. Then we derived the Newton linearisation of the two problems and the corresponding algorithms. We tested the problem using the Method of Manufactured Solutions (MMS). To ensure the compatibility of the algorithms with the less smooth problem, we investigated different types of globalisation of Newton's method, such as backtracking line search and critical point line search. Finally, we performed empirical complexity analysis of each methods and tabulated the evaluation counts.

1 Introduction

Solving nonlinear partial differential equations describing the deformation of solid structures is a ubiquitous problem in many areas of engineering. For example, in the field of aeroplane design, the optimisation of the shape of wings and the prediction of their behaviour under heavy loads plays a key role for their safety, economic efficiency of manufacturing, and fuel efficiency. A commonly simulated problem is the deformation of the wing, modelled by nonlinear elasticity. This project seeks to study and improve Newton's method for the approximate solution of the underlying PDEs.

The Newton–Kantorovich iteration, the most widely used method for the numerical solution of nonlinear PDEs, generalises the Newton–Raphson method (for rootfinding of scalar functions) to infinite-dimensional normed vector spaces. This will be combined with the finite element method (FEM), which provides approximate solutions to the unknowns in terms of piecewise polynomials, to generate computer simulations of some problems motivated by concrete real-world applications.

In particular, we will adopt the high-order conforming finite element methods which have become increasingly popular due to their increased accuracy, and compatibility with modern supercomputers, and have also been shown to avoid the phenomenon of *locking* in the case of linear elasticity.

We will study the abstract theory of Newton's method in Banach spaces and the FEM, and code numerical experiments (in the finite element library Firedrake [4]) for solid mechanics problems to give results with which to compare the theory.

2 Introduction to the elasticity equations

2.1 Linear elasticity model

Given an elastic body occupying the domain $\Omega \subseteq \mathbb{R}^2$ with boundary $\partial\Omega = \Gamma_1 \sqcup \Gamma_2$, we seek a displacement field $u : \Omega \to \mathbb{R}^2$. The linear elasticity equation can be derived from minimising the Hookean energy, defined

*Department of Computational Applied Mathematics and Operations Research, Rice University, 6100 Main St. - MS 134, Houston, TX 77005 (nick.nie@rice.edu).

†Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556 (faznaran@nd.edu).

‡Mathematical Institute, University of Oxford, Oxford, OX2 6GG, UK (parker@maths.ox.ac.uk).

by

$$J(u) := \int_\Omega \mu|\varepsilon(u)|^2 + \frac{1}{2}\lambda|\nabla\cdot u|^2 - f\cdot u \,\mathrm{d}x - \int_{\Gamma_2} g\cdot u \,\mathrm{d}s, \tag{2.1}$$

for $u - u_0 \in V := H^1_{\Gamma_1}(\Omega;\mathbb{R}^2) := \{v \in H^1(\Omega;\mathbb{R}^2) \mid v|_{\Gamma_1} = 0\}$, where $u_0 : \Omega \to \mathbb{R}^2$ is the so-called prescribed configuration, giving the displacement of the body on the boundary Γ_1. Here, $f : \Omega \to \mathbb{R}^2$ is a body force, $g : \Omega \to \mathbb{R}^2$ is a prescribed surface traction, $\varepsilon(u) := \frac{1}{2}(\nabla u + (\nabla u)^\top)$ is the symmetric gradient and $\mu, \lambda > 0$ are the so-called Lamé parameters [2].

Note that we take the convention that gradient of a vector field gives a matrix whose row is the gradient of the each element of the vector field and divergence of a matrix field is defined row-wise. This problem is motivated by the modelling of a bridge bending under the force of gravity.

To find a minimiser of the energy functional, we look for the stationary point of the functional instead. This can be achieved by finding its directional derivative $J'(u;v)$ where $v \in V$.

Definition 2.1. (Directional differentiability.) Let $J : V \to W$, where V and W are Banach spaces. The directional derivative of J evaluated at $u \in V$ in the direction $v \in V$ is

$$J'(u;v) = \lim_{t\to 0^+} \frac{J(u+tv) - J(u)}{t}, \tag{2.2}$$

if the limit exists.

In this case, we have

$$\begin{aligned}
J'(u;v) &:= \lim_{t\to 0^+} \frac{J(u+tv) - J(u)}{t} \\
&= \lim_{t\to 0^+} \frac{1}{t}\left(\int_\Omega \mu|\varepsilon(u+tv)|^2 + \frac{1}{2}\lambda|\nabla\cdot(u+tv)|^2 - f\cdot(u+tv)\,\mathrm{d}x \right. \\
&\quad \left. - \int_{\Gamma_2} g\cdot(u+tv)\,\mathrm{d}s - \int_\Omega \mu|\varepsilon(u)|^2 + \frac{1}{2}\lambda|\nabla\cdot u|^2 - f\cdot u\,\mathrm{d}x + \int_{\Gamma_2} g\cdot u\,\mathrm{d}s\right) \\
&= \int_\Omega 2\mu\varepsilon(u) : \varepsilon(v) + \lambda(\nabla\cdot u)(\nabla\cdot v)\,\mathrm{d}x - \int_\Omega f\cdot v\,\mathrm{d}x - \int_{\Gamma_2} g\cdot v\,\mathrm{d}s.
\end{aligned}$$

Setting $J'(u;v) = 0$ gives the weak form of the linear elasticity equation:

$$\int_\Omega 2\mu\varepsilon(u) : \varepsilon(v) + \lambda(\nabla\cdot u)(\nabla\cdot v)\,\mathrm{d}x = \int_\Omega f\cdot v\,\mathrm{d}x + \int_{\Gamma_2} g\cdot v\,\mathrm{d}s \;\forall v \in V. \tag{2.3}$$

If we assume that the solution is twice continuously differentiable instead, then we can derive the following strong form of the equation.

Standard form:

$$-2\mu\,\nabla\cdot\varepsilon(u) - \lambda\nabla\,\nabla\cdot u = f \quad \text{in } \Omega, \qquad (\textit{conservation of linear momentum}) \tag{2.4a}$$

$$u = u_0 \quad \text{on } \Gamma_1, \qquad (\textit{prescribed displacement}) \tag{2.4b}$$

$$(2\mu\varepsilon(u) + \lambda(\nabla\cdot u)\mathbb{I})\,\mathbf{n} = g \quad \text{on } \Gamma_2, \qquad (\textit{prescribed surface traction}) \tag{2.4c}$$

$\mathbf{n}$ is the unit normal to $\partial\Omega$, $\mathbb{I}$ is the identity matrix.

2.2 SVK model

The SVK hyperelasticity model can be seen as a variant of the linear elasticity model where the strain tensor, instead of being linear in the gradient of the displacement, is now quadratic [2].

$$\boldsymbol{E}_{SVK} = \frac{1}{2}(\nabla u + \nabla u^\top + (\nabla u)^\top \nabla u). \tag{2.5}$$

On the other hand, the constitutive equation, describing the stress-strain relationship, is still linear, identical to the linear elasticity case [2]. The following is the the stored energy function

$$\check{W}(\boldsymbol{E}_{SVK}) = \frac{\lambda}{2}(\operatorname{tr} \boldsymbol{E}_{SVK})^2 + \mu \operatorname{tr}(\boldsymbol{E}_{SVK}^2). \tag{2.6}$$

As such, we have the following weak formulation of the SVK hyperelasticity equation:

$$\int_\Omega \left(\mu(\mathbb{F}(u)^\top \mathbb{F}(u) - \mathbb{I}) + \lambda \left(\nabla\cdot u + \frac{|\nabla u|^2}{2}\right)\mathbb{F}(u)\right) : \nabla v \, \mathrm{d}x = \int_\Omega f \cdot v \, \mathrm{d}x + \int_{\Gamma_2} g \cdot v \, \mathrm{d}s \;\forall\; v \in V, \tag{2.7}$$

where $\mathbb{F}(u) := \mathbb{I} + \nabla u$ is the deformation gradient.

3 Introduction to Newton's method

We will introduce the key definition of the report in this section, the Newton–Kantorovich iteration for the solution of the abstract rootfinding problem

$$F(u) = 0 \tag{3.1}$$

where $F : V \to W$ for Banach spaces V, W. Typically in the context of PDEs, we have that W is a Sobolev space, and $W = V^*$.

Here are the problems we would like to solve in this nonlinear form.

Linear elasticity:

$$F_L(u)(v) = \int_\Omega 2\mu\varepsilon(u) : \varepsilon(v) + \lambda(\nabla\cdot u)(\nabla\cdot v) \, \mathrm{d}x \tag{3.2}$$

$$- \int_\Omega f \cdot v \, \mathrm{d}x - \int_{\Gamma_2} g \cdot v \, \mathrm{d}s = 0 \quad \forall v \in H^1_{\Gamma_1}(\Omega;\mathbb{R}^2). \tag{3.3}$$

SVK hyperelasticity:

$$\begin{aligned} F_{SVK}(u)(v) = &\int_\Omega \left(\mu(\mathbb{F}(u)^\top \mathbb{F}(u) - \mathbb{I}) + \lambda \left(\nabla\cdot u + \frac{|\nabla u|^2}{2}\right)\mathbb{F}(u)\right) : \nabla v \, \mathrm{d}x \\ &- \int_\Omega f \cdot v \, \mathrm{d}x - \int_{\Gamma_2} g \cdot v \, \mathrm{d}s = 0 \quad \forall v \in W^{1,4}_{\Gamma_1}(\Omega;\mathbb{R}^2). \end{aligned} \tag{3.4}$$

Before introducing the algorithm, we need a few key definitions from functional analysis.

Definition 3.1. (Gâteaux differentiability.) If J is directionally differentiable at u, and there exists a bounded linear map $J'(u) : V \to W$ such that

$$J'(u; v) = J'(u)v, \tag{3.5}$$

then J is Gâteaux differentiable at u with derivative $J'(u)$.

Definition 3.2. (Fréchet differentiability.) Suppose $J : V \to W$ is Gâteaux differentiable at a point $u \in V$ and that the derivative J' satisfies

$$\lim_{v\to 0} \frac{\|J(u+v) - J(u) - J'(u)v\|_W}{\|v\|_V} = 0 \quad \text{for all } v \in V. \tag{3.6}$$

Then J is Fréchet differentiable at u.

Now we are ready to present the Newton–Kantorovich algorithm for PDEs. Typically, V is a Sobolev space and $W = V^*$. Suppose that $F : V \to V^*$ is Fréchet differentiable everywhere between Banach spaces V, V^*. Then given $u_0 \in V$, the Newton–Kantorovich iteration is

$$u^{k+1} := u^k - F'(u^k)^{-1}F(u^k), \quad k \geq 0, \tag{3.7}$$

where $F'(u) : V \to V^*$ is the Fréchet derivative (Jacobian) of F at $u \in V$.

Rearranging (3.7) and defining $\delta u^k := u^{k+1} - u^k$, we get

$$F'(u^k)\delta u^k = -F(u^k). \tag{3.8}$$

Since F is also Gâteaux differentiable, we have $F'(u^k)\delta u^k = F'(u^k; \delta u^k)$. If we choose some test function $v \in V$, we get the following Newton iterates for PDEs:

$$G_u(u^k; v, \delta u^k) = -G(u^k; v) \quad \forall v \in V, \tag{3.9}$$

where $G(u; v) := F(u)(v)$ and G_u is the directional derivative of G. Then we have the following update:

$$u^{k+1} = u^k + \delta u^k. \tag{3.10}$$

To write this formally, we formulate the solver as follows.

Algorithm 1 Classical Newton's Method

procedure ClassicalNewton(G, u^0)
  Choose $k_{\max} \in \mathbb{Z}^{>0}$
  for $1 \leq k \leq k_{\max}$ **do**
    Find δu^k s.t. $G_u(u^k; v, \delta u^k) = -G(u^k; v) \quad \forall v \in V$
    if stopping criterion holds **then**
      Break
    end if
    $u^{k+1} \leftarrow u^k + \delta u^k$
  end for
  return u^k
end procedure

With the above algorithm, we also need to implement specific stopping criteria to determine when to terminate the algorithm and whether it converges. Suppose that the exact solution to the equation is u^*. One naive criterion for convergence is simply the norm of the error $\|u^k - u^*\| < \texttt{tol}$ for some tolerance `tol`. However, since we do not have access to the exact solution u^*, this term can never be evaluated. Instead, we can consider the norm of the residual $\|F(u^k) - F(u^*)\| = \|F(u^k)\|$ by leveraging the fact that $F(u^*) = 0 \in V^*$.

Note, however, that $F(u^k) \in V^*$. In our case, we want to minimise the norm of the residual $\|F(u)\|_V$ in some norm. We may choose the norm as follows $\|F(u)\|_V = \|(G(u; v_1), \ldots, G(u; v_n))\|_{\ell^2}$ where $\{v_i\}_{i=1}^n$ is a basis for the finite element subspace $V_h \subset V$.

4 Introduction to Galerkin Approximation

Suppose that we have a bilinear form $a(u; v)$ and a bounded linear functional $L(v)$ where $a : V \times V \to \mathbb{R}$ and $L : V \to \mathbb{R}$. Then our variational problem can be written as

$$\text{Find } u \in V \text{ s.t. } a(u; v) = L(v) \quad \forall v \in V. \tag{4.1}$$

Note that typically V is an infinite dimensional function space so finding the exact $u \in V$ is often intractable. One common solution is to use piecewise continuous polynomials to approximate u. Suppose that we have the following set of linearly independent piecewise continuous polynomials $\{v_i\}_{i=1}^n$. Then we can define the finite dimensional subspace $V_h \subset V$ of the original space by $V_h = \text{span}(v_1, \ldots, v_n)$. Now suppose that our approximate solution $u_h \in V_h$ takes the form $u_h(x) = \sum_{i=1}^n \gamma_i v_i(x)$ where $\gamma_i \in \mathbb{R}$. If we restrict our test function to V_h, we have the following Galerkin approximation of the original variational problem

$$\text{Find } u_h \in V_h \text{ s.t. } a(u_h; v_h) = L(v_h) \quad \forall v_h \in V_h. \tag{4.2}$$

5 Linearisation of the elasticity equations

From the previous section, we note that in order to apply Newton's method, we need to linearise the equation. This involves computing the derivative of $F(u)$ where the original equation is of the form $F(u) = 0$. Here, $F(u)$ is a bounded linear functional on the space V. Equivalently, fixing a particular $v \in V$, it suffices to find the derivative of $G(u; v)$. We will consider the linear elasticity model and the SVK model separately.

5.1 Linear elasticity

Note that the linear elasticity equation can be rewritten in the general form $G(u; v) = 0$, where

$$G(u; v) := \int_\Omega 2\mu\varepsilon(u) : \varepsilon(v) + \lambda(\nabla\cdot u)(\nabla\cdot v)\ \mathrm{d}x - \int_\Omega f\cdot v\ \mathrm{d}x - \int_{\Gamma_2} g\cdot v\ \mathrm{d}s. \tag{5.1}$$

Consider

$$G(u + tw; v) = \int_\Omega 2\mu\varepsilon(u + tw) : \varepsilon(v) + \lambda(\nabla\cdot(u + tw))(\nabla\cdot v)\ \mathrm{d}x - \int_\Omega f\cdot v\ \mathrm{d}x - \int_{\Gamma_2} g\cdot v\ \mathrm{d}s. \tag{5.2}$$

Since $\varepsilon(u + tw) = \varepsilon(u) + t\varepsilon(w)$ and $\nabla\cdot(u + tw) = \nabla\cdot u + t\,\nabla\cdot w$, then

$$G_u(u; v, w) := \lim_{t\to 0} \frac{G(u + tw; v) - G(u; v)}{t} = \int_\Omega 2\mu\varepsilon(w) : \varepsilon(v) + \lambda(\nabla\cdot w)(\nabla\cdot v)\ \mathrm{d}x. \tag{5.3}$$

If we define the Cauchy stress as $\sigma(u) := 2\mu\varepsilon(u)+\lambda(\nabla\cdot u)\mathbb{I}$, then the Newton update δu^k satisfies $G_u(u^k; v, \delta u^k) = -G(u^k; v)$ which can be written as

$$\int_\Omega \sigma(\delta u^k) : \varepsilon(v)\ \mathrm{d}x = -\int_\Omega \sigma(u^k) : \varepsilon(v)\ \mathrm{d}x + \int_\Omega f\cdot v\ \mathrm{d}x + \int_{\Gamma_2} g\cdot v\ \mathrm{d}s. \tag{5.4}$$

Note that at the first iteration, we have

$$\int_\Omega \sigma(\delta u^0) : \varepsilon(v)\ \mathrm{d}x = -\int_\Omega \sigma(u^0) : \varepsilon(v)\ \mathrm{d}x + \int_\Omega f\cdot v\ \mathrm{d}x + \int_{\Gamma_2} g\cdot v\ \mathrm{d}s. \tag{5.5}$$

Rearranging,

$$\int_\Omega \sigma(\delta u^0) : \varepsilon(v)\ \mathrm{d}x + \int_\Omega \sigma(u^0) : \varepsilon(v)\ \mathrm{d}x - \int_\Omega f\cdot v\ \mathrm{d}x - \int_{\Gamma_2} g\cdot v\ \mathrm{d}s = 0. \tag{5.6}$$

$$\int_\Omega \sigma(u^0 + \delta u^0) : \varepsilon(v)\ \mathrm{d}x - \int_\Omega f\cdot v\ \mathrm{d}x - \int_{\Gamma_2} g\cdot v\ \mathrm{d}s = 0. \tag{5.7}$$

Then $u^0 + \delta u^0$ solves the original PDE, i.e. Newton's method converges in one iteration. This is consistent with the observation that the original elasticity PDE is linear in u.

5.2 SVK hyperelasticity

The SVK equation is nonlinear so we expect the linearisations to differ from the original equation. The operator is defined as follows:

$$\begin{aligned} G(u; v) = &\int_\Omega \left(\mu(\mathbb{F}(u)^\top\mathbb{F}(u) - \mathbb{I}) + \lambda\left(\nabla\cdot u + \frac{|\nabla u|^2}{2}\right)\mathbb{F}(u)\right) : \nabla v\ \mathrm{d}x \\ &- \int_\Omega f\cdot v\ \mathrm{d}x - \int_{\Gamma_2} g\cdot v\ \mathrm{d}s\ \forall v \in W^{1,4}_{\Gamma_1}(\Omega; \mathbb{R}^2), \end{aligned} \tag{5.8}$$

where $\mathbb{F}(u) = \mathbb{I} + \nabla u$ is the deformation gradient. Consider $G(u + tw; v)$, we get

$$G(u+tw;v) = \int_\Omega \left(\mu(\mathbb{F}(u+tw)^\top \mathbb{F}(u+tw) - \mathbb{I}) + \lambda \left(\nabla\cdot(u+tw) + \frac{|\nabla(u+tw)|^2}{2}\right)\mathbb{F}(u+tw)\right) : \nabla v \ \mathrm{d}x \\ - \int_\Omega f\cdot v \ \mathrm{d}x - \int_{\Gamma_2} g\cdot v \ \mathrm{d}s. \tag{5.9}$$

Now we have $\mathbb{F}(u + tw) = \mathbb{F}(u) + t\nabla w$. Hence,

$$\mathbb{F}(u+tw)^\top \mathbb{F}(u+tw) = \mathbb{F}(u)^\top \mathbb{F}(u) + t(\mathbb{F}(u+tw)^\top \nabla w + \nabla w^\top \mathbb{F}(u+tw)) + \mathcal{O}(t^2). \tag{5.10}$$

We can also expand $|\nabla(u + tw)|^2 = |\nabla u + t\nabla w|^2 = |\nabla u|^2 + 2t\nabla u : \nabla w + t^2|\nabla w|^2$.

$$G_u(u;v,w) = \int_\Omega \Bigg(\mu\left(\mathbb{F}(u)^\top \nabla w + \nabla w^\top \mathbb{F}(u)\right) \\ + \lambda\left((\nabla\cdot w + \nabla u : \nabla w)\mathbb{F}(u) + \left(\nabla\cdot u + \frac{|\nabla u|^2}{2}\right)\nabla w\right)\Bigg) : \nabla v \ \mathrm{d}x \\ - \int_\Omega f\cdot v \ \mathrm{d}x - \int_{\Gamma_2} g\cdot v \ \mathrm{d}s. \tag{5.11}$$

6 Globalisation of Newton's method

The primary methods for globalisation in this project are the line search method and the load continuation method. The line search method is a method of choosing a step size β such that the update $u^{k+1} := u^k + \beta\delta u^k$ gives a better convergence property than the unscaled update. While line search is a very generic method, load continuation is rather specialised for solid mechanics. The load continuation method refers to solving a sequence of problems with forcing functions f and g, also called load, of increasing magnitude, and using the solution to the previous problem as the initial guess for the Newton solver for the next problem in the sequence. This ensures that the solutions in the sequence are close to each other so that final configuration is a continuous deformation from the original configuration when continuously increasing force is applied.

6.1 Line search method

As for line search methods, we mainly implemented two methods: the backtracking line search and the critical point line search.

6.1.1 Backtracking line search

The backtracking line search method refers to an algorithm that scales the step size by a factor of $\tau \in (0, 1)$ when a certain descent condition is violated [5]. Typically, $\tau = 0.5$. Let h be a function we want to minimise.

One commonly used descent condition is the Armijo condition [5] given by the following

$$h(x^k + \beta\delta x^k) \leq h(x^k) + c\beta\nabla h(x^k)^\top \delta x^k \tag{6.1}$$

where c is some small constant, e.g. $c = 0.1$. While this condition usually gives rise to algorithm with good performance, the derivative term in the condition is rather costly to compute. Instead, using the idea from the Armijo strategy [3], we obtain the following derivative-free descent condition

$$h(x^k + \beta\delta x^k) \leq (1 - c\beta)h(x^k). \tag{6.2}$$

In our case, we want to minimise the norm of the residual $\|F(u)\|_V$ in some norm. We may choose the norm as follows $\|F(u)\|_V = \|(G(u; v_1), \ldots, G(u; v_n))\|_{\ell^2}$ where $\{v_i\}_{i=1}^n$ is a basis for the finite element subspace $V_h \subset V$. Hence, we can obtain the following backtracking line search algorithm.

Algorithm 2 Backtracking line search

Choose $\beta = 1, \tau \in (0,1), i_{\max} \in \mathbb{Z}^{>0}$
for $1 \leq i \leq i_{\max}$ **do**
  if $\|F(u^k + \beta\delta u^k)\|_V \geq (1 - c\beta)\|F(u^k)\|_V$ **then**
    $\beta \leftarrow \tau\beta$
  else
    Break
  end if
end for
return β

6.1.2 Critical point line search

The critical point method is a specific line search method for PDEs with an energy structure [1]. It assumes that the solution to the PDE is a critical point of an energy functional. While the method replies on the energy associated with the PDE, it does not need it to be computed explicitly. Hence, it is still useful for PDEs with an energy that is challenging to compute.

The construction of the method is as follows. Suppose that there is an energy functional $J(v)$ such that the solution u of the PDE $F(u) = 0$ is a stationary point of $J(v)$. Given the current iterate u and the update δu, we would like to find a β such that $\frac{J(u+\beta\delta u)}{d\beta} = G(u + \beta\delta u; \delta u) = 0$, i.e. finding a stationary point of $J(u)$ in the direction δu. One may hope that this stationary point will minimise the energy functional.

One way to find such a β is to use Newton's method for the scalar equation. We can either use the secant method to estimate the derivative of G [1] , a quasi-Newton-type method, or use the exact Newton method. These give rise to two slightly different updates. The two algorithms are given as follows:

Algorithm 3 CPLS using Quasi-Newton

Choose $\beta_0 = 0, \beta_1 = 1, i_{\max} \in \mathbb{Z}^{>0}$.
for $1 \leq i \leq i_{\max}$ **do**
  if $|\beta_i - \beta_{i-1}| >$ tol **then**
    $\beta_{i+1} \leftarrow \beta_i - \frac{G(u+\beta_i\delta u;\delta u)(\beta_i - \beta_{i-1})}{G(u+\beta_i\delta u;\delta u) - G(u+\beta_{i-1}\delta u;\delta u)}$
  else
    Break
  end if
end for
return β_i

Algorithm 4 CPLS using exact Newton

Choose $\beta_0 = 0, \beta_1 = 1, i_{\max} \in \mathbb{Z}^{>0}$.
for $1 \leq i \leq i_{\max}$ **do**
  if $|\beta_i - \beta_{i-1}| >$ tol **then**
    $\beta_{i+1} \leftarrow \beta_i - G_u(u + \beta_i\delta u; \delta u, \delta u)^{-1} G(u + \beta_i\delta u; \delta u)(\beta_i - \beta_{i-1})$
  else
    Break
  end if
end for
return β_i

One thing to note in practice is that the step size can be numerically unstable in these algorithms, especially when the values of $G(u+\delta u; \delta u)$ and $G(u; \delta u)$ are very close to each other. In this case, they might become very large or even negative. One way to get over this is to regularise the step size by choosing the

maximum and the minimum step size. This effectively prevents the algorithm from diverging. However, the exact values of the regularisation need to be tuned for each problem carefully.

6.2 Load continuation method

Load continuation is a special application of the method of parameter continuation to solid mechanics [3]. It is also named the "parametric force increment" in the reference [6]. The parameter continuation method refers to a class of method for solving PDEs with a parameter by solving a sequence of problem with increasing parameters and using the solution to the previous problem as the initial guess for the next problem [3]. The load continuation method introduces an artificial parameter α ranging from 0 to 1 applied to the load f and g, where 0 represents no load and 1 represents the full load. Then we apply the load continuation method with respect to the parameter α. For hyperelastic materials, the deformation is fully reversible when the stress is removed with no plastic deformation [6]. This gives the theoretical justification for load continuation since the final configuration when the full load is applied is a continuous deformation from the original configuration as the load is progressively increased.

Recall that the original SVK problem is given as follows

$$a(u;v) := \int_\Omega \left(\mu(\mathbb{F}(u)^\top \mathbb{F}(u) - \mathbb{I}) + \lambda\left(\nabla\cdot u + \frac{|\nabla u|^2}{2}\right)\mathbb{F}(u)\right) : \nabla v \, \mathrm{d}x \tag{6.3}$$

$$L(v) := \int_\Omega f\cdot v \, \mathrm{d}x + \int_{\Gamma_2} g\cdot v \, \mathrm{d}s \tag{6.4}$$

$$a(u;v) = L(v) \;\forall\; v \in V. \tag{6.5}$$

To implement load continuation, we need to generate an increasing sequence of parameters $0 = \alpha_0 \leq \alpha_1 \leq \cdots \leq \alpha_N = 1$. We can solve for $u_{(i)}$ which is the solution to the equation $a(u_{(i)};v) = \alpha_i L(v)$. Then we use $u_{(i)}$ as the initial guess for Newton's method applied to $a(u_{(i+1)};v) = \alpha_{i+1}L(v)$ and in turn solve for $u_{(i+1)}$.

For the load continuation method, we will use a uniform continuation in the SVK model, i.e. $\alpha_i - \alpha_{i-1} = \frac{1}{N}$.

Algorithm 5 Load continuation

Choose $N \in \mathbb{Z}^{>0}$
for $1 \leq n \leq N$ **do**
  $G_n(u;v) := a(u;v) - \frac{n}{N}L(v)$
  $u_n \leftarrow$ ClassicalNewton(G_n, u_{n-1})
end for
return u_N

7 Computational experiments

We will now carry out computational experiments using algorithms from the previous section on the SVK model. To compare different methods, we can use the number of bilinear form and bounded linear functional evaluations and the number of linear solvers called as the complexity measures since they are the most costly steps in the solver. We will use N_eval to represent the number of bilinear form and bounded linear functional evaluations and N_solve the number of linear solvers called.

Now we will tabulate the complexity comparison for different methods. We fix the parameters as follows: $\mu = 1, \lambda = 1$, levels = 3, N_line = 3 (, N_param = 3 if load continuation used).

The number marked with an asterisk is the one that reaches the maximum number of iterations but still has not satisfied the stopping criterion. Hence, these methods are at least as costly as the number indicated. Note that the classical Newton method is the least expensive method. The backtracking line search and the load continuation method are both reasonably fast to run. The two types of Newton line search method, quasi-Newton or exact Newton, not only reach the maximum number of iterations but are also noticeably slower to run in Firedrake. In addition, the backtracking line search method gives a small residual among

Table 1: Method comparisons for fixed parameters

Method	**Classical Newton**	**Backtracking LS**	**Quasi-Newton LS**	**Exact Newton LS**	**Load continuation**
N_eval	46	114	*126	*122	104
N_solve	20	24	*19	*19	43

Table 2: Method comparisons for fixed parameters (continued)

Method	**Load continuation with backtracking LS**	**Load continuation with quasi-Newton LS**	**Load continuation with exact Newton LS**
N_eval	212	*246	*224
N_solve	48	*39	*39

all the iterates. Hence, we are more confident that this method will converge to the physical solution when multiple solutions are present.

Note that not all parameters λ and κ will give a convergent algorithm. When κ is larger, the deformation observed is larger and eventually results in a singularity of the domain at one of the corners of the mesh.

We will summarise the approximate permitted parameter range in the following table.

Table 3: Convergent parameter regime for square mesh(1,1), levels = 3

λ	1	10	10^2	10^3	10^4	10^5
κ	≤ 0.1	$\leq 10^{-2}$	$\leq 10^{-3}$	$\leq 10^{-4}$	$\leq 10^{-6}$	$\leq 10^{-6}$

Within the permitted parameter range, if κ is chosen to be very small, the algorithm will converge in very few steps. Hence, these choices of the parameters would not be very useful in comparing the computational complexity for different sizes of the parameters. One way to get around with this is to consider the worst case scenario, choosing the largest possible κ for each λ. The summary of complexity measures for each method with different parameters is given as follows.

Table 4: Complexity analysis for classical Newton

λ	1	10	10^2	10^3	10^4	10^5
κ	0.1	10^{-2}	10^{-3}	10^{-4}	10^{-6}	10^{-6}
N_eval	46	34	30	28	26	*118
N_solve	20	14	12	11	10	*56

Note that as λ becomes larger, the computational cost decreases. However, with the given choice of the tolerance, the algorithms associated with last one or two parameter choices do not converge. Examining the residuals of the these algorithms, we notice that the residuals quickly get small but stagnate at values above the tolerance. One possible explanation for this is that the linear system involved in the problem becomes more and more ill-conditioned as the parameter λ gets larger. Hence, the error in the solver itself may exceed the error of the iterates from the solution. This does not mean that the algorithm diverges but it means a more relaxed tolerance should be chosen as a measure of convergence. The following tables are constructed using a relaxed tolerance.

To illustrate the differences in costs, we can plot the two complexity measures in two separate plots.

Table 5: Complexity analysis for Newton with backtracking LS

λ	1	10	10^2	10^3	10^4	10^5
κ	0.1	10^{-2}	10^{-3}	10^{-4}	10^{-6}	10^{-6}
N_eval	114	70	60	52	52	*234
N_solve	24	16	14	12	12	*57

Table 6: Complexity analysis for Newton with load continuation

λ	1	10	10^2	10^3	10^4	10^5
κ	0.1	10^{-2}	10^{-3}	10^{-4}	10^{-6}	10^{-6}
N_eval	104	82	76	74	*166	*166
N_solve	43	32	29	28	*74	*74

Table 7: Complexity analysis for classical Newton with relaxed tolerance

λ	1	10	10^2	10^3	10^4	10^5
κ	0.1	10^{-2}	10^{-3}	10^{-4}	10^{-6}	10^{-6}
N_eval	44	32	28	26	24	24
N_solve	19	13	11	10	9	9

Table 8: Complexity analysis for Newton with backtracking LS with relaxed tolerance

λ	1	10	10^2	10^3	10^4	10^5
κ	0.1	10^{-2}	10^{-3}	10^{-4}	10^{-6}	10^{-6}
N_eval	110	66	52	48	48	48
N_solve	23	15	12	11	11	11

Table 9: Complexity analysis for Newton with load continuation with relaxed tolerance

λ	1	10	10^2	10^3	10^4	10^5
κ	0.1	10^{-2}	10^{-3}	10^{-4}	10^{-6}	10^{-6}
N_eval	100	80	74	72	72	72
N_solve	41	31	28	27	27	27

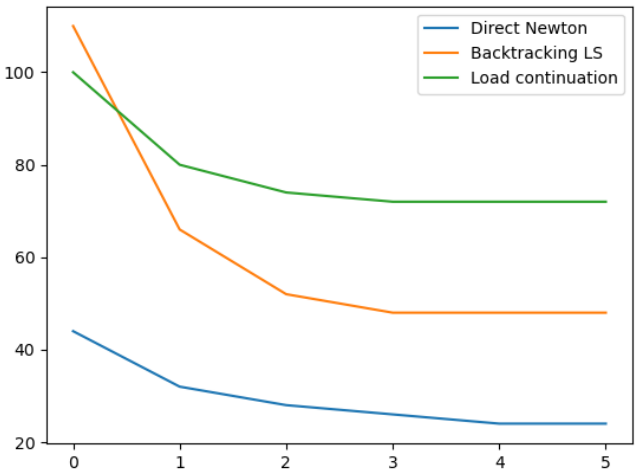

Figure 1: N_eval for different methods

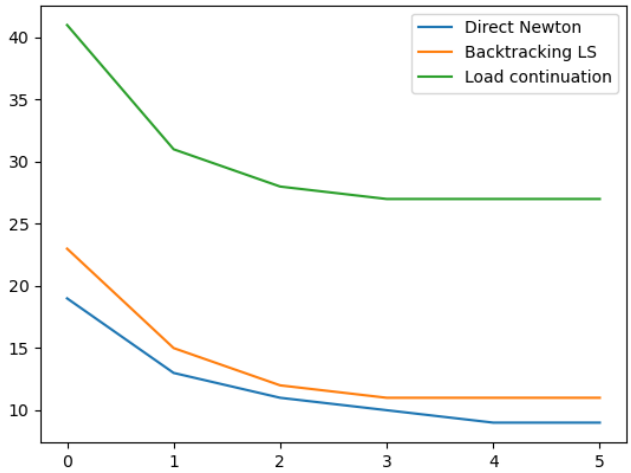

Figure 2: N_solve for different methods

From the plot, we can see that the classical Newton method is still the least costly one. The other two methods are both much more costly. From the table, we can see that as λ gets larger, the load continuation method is almost three times as costly as classical Newton. This is consistent with our construction since the load continuation solves a sequence of three problems in total and each sub-problem is close in computational cost to the full problem using classical Newton. From the output of the method, the residuals of the iterates are the most stable for the backtracking line search method.

While the data show that the classical Newton method works the best for this test problem, the other two methods have their own place. The backtracking line search improves the stability of the algorithm and the load continuation method is very useful when the parameter gets close to the problematic regime. In practice, for more complicated stress functions, it is better to combine the backtracking line search and the load continuation method to guarantee convergence.

8 Conclusion

In this work, we explored the application of Newton's method to the elasticity equations. We first derived the linear elasticity equation by minimising the Hookean energy functional. The nonlinear version, the so-called SVK hyperelasticity, is derived from a nonlinear coordinate transformation of the linear elasticity case. Then we introduced the Newton–Kantorovich iteration, or simply Newton's method, for PDEs. In order to apply Newton's method, we linearised the PDE. This gives the corresponding Newton's iterations. While Newton's method is quadratically convergent when the initial guess is sufficiently close to the solution, in reality, this is almost never the case since we are considering a case with large load applied. Hence, we considered two types of globalisation techniques: line search and load continuation. We derived the corresponding algorithms for these two modifications to the classical Newton method. After this, we ran computational experiments involving a square mesh with a fixed stress applied at $x = 1$. After some comparisons, we noticed the classical Newton's method is the least costly algorithm for the proposed problem. However, the backtracking line search and the load continuation methods are also relatively efficient and robust to large stress. Depending on the choice for f and g in the PDE, the globalisation techniques should be chosen accordingly.

9 Future work

In this work, when solving the SVK problem, we implicitly made the assumption that the material cannot self-intersect. More formally, this requires the assumption that $\det \nabla u > 0$ a.e. in Ω.

It turns out that the explicitly imposing this constraint is essential to ensure that the solution we obtain is physical. In our case, when the mesh is rather coarse, the solution does not exhibit self-intersection. However, when the mesh is refined, Newton's method fails to converge due to the singularity of the domain. In order to resolve the issue, we can encode this inequality constraint into the energy functional, or correct self-intersecting solutions during Newton's iterates.

There are also alternative linearisations and discretisations of the PDEs that are worth exploring. For instance, we can consider Picard linearisation, or minimisation of the strain energy via higher-order Taylor expansions that the linear approximation on which Newton is based. As for discretisations, we have limited our investigations to the continuous Galerkin method. We may consider nonconforming elements, the discontinuous Galerkin method, use a higher-order polynomial for the element. These will be left for future investigations.

References

[1] P. R. Brune, M. G. Knepley, B. F. Smith, and X. Tu, *Composing scalable nonlinear algebraic solvers*, SIAM Review, 2015, pp. 535–565, https://doi.org/10.1137/130936725. 7

[2] P. G. Ciarlet, *Mathematical elasticity Volume I: Three-dimensional elasticity*, SIAM, Philadelphia, 1988, https://doi.org/10.1137/1.9781611976786. 2, 3

[3] P. Deuflhard, *Newton methods for nonlinear problems: Affine invariance and adaptive algorithms*, vol. 35, Springer Science & Business Media, 2005, https://doi.org/10.1007/978-3-642-23899-4. 6, 8

[4] D. A. Ham, P. H. J. Kelly, L. Mitchell, C. J. Cotter, R. C. Kirby, K. Sagiyama, N. Bouziani, S. Vorderwuelbecke, T. J. Gregory, J. Betteridge, D. R. Shapero, R. W. Nixon-Hill, C. J. Ward, P. E. Farrell, P. D. Brubeck Martinez, I. Marsden, T. H. Gibson, M. Homolya, T. Sun, A. T. T. McRae, F. Luporini, A. Gregory, M. Lange, S. W. Funke, F. Rathgeber, G. T. Bercea, and G. R. Markall, *Firedrake user manual*, Imperial College London and University of Oxford and Baylor University and University of Washington, 1st ed., May 2023, https://doi.org/10.25561/104839. 1

[5] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer, New York, 2nd ed., 2006, https://doi.org/10.1007/978-0-387-40065-5. 6

[6] Wolfram Research Inc., *Hyperelasticity–Wolfram Language Documentation*, 2023. https://reference.wolfram.com/language/PDEModels/tutorial/StructuralMechanics/Hyperelasticity.html. Accessed 19th August 2023. 8