12^{th} World Congress on Structural and Multidisciplinary Optimization
5^{th} - 9^{th}, June 2017, Braunschweig Germany

Efficient Optimal Surface Texture Design Using Linearization

Chendi Lin[1], Yong Hoon Lee[1], Jonathon K. Schuh[1], Randy H. Ewoldt[1], James T. Allison[2]

[1] Deparment of Mechanical Science and Engineering
[2] Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign, Urbana IL USA, clin92@illinois.edu

1. Abstract

Surface textures reduce friction in lubricated sliding contact. This behavior can be modeled using the Reynolds equation, a single partial differential equation (PDE) that relates the hydrodynamic pressure to the gap height. In a previous study, a free-form texture design optimization problem was solved based on this model and two competing design objectives. A pseudo-spectral method was used for PDE solution, which was treated as a black box in the optimization problem. This optimization implementation did not exploit model structure to improve numerical efficiency, so design representation fidelity was limited. Here a new strategy is introduced where design representation resolution and computational efficiency are improved simultaneously. This is achieved by introducing a new optimization variable involving both pressure gradient and the cube of gap height at each mesh node location, and simultaneously solving the flow and texture design problems. This transformation supports linearization of the governing equations and design objectives. Sequential Linear Programming (SLP) is used with the epsilon-constraint method to generate Pareto-optimal texture designs with high resolution and low computational expense. An adaptive trust region is used, based on solution improvement, to manage linearization error. Comparing to the non-linear programming implementation, the solution converged to a set of slightly suboptimal points (maximum 25% objective function degradation when normalized apparent viscosity is 0.5, and moderately better when normalized apparent viscosity is 0.2), but results in significant improvement in computational speed (8.4 times faster).

2. Keywords: Multi-objective optimization, Linearization, Shape optimization, Reynolds equation

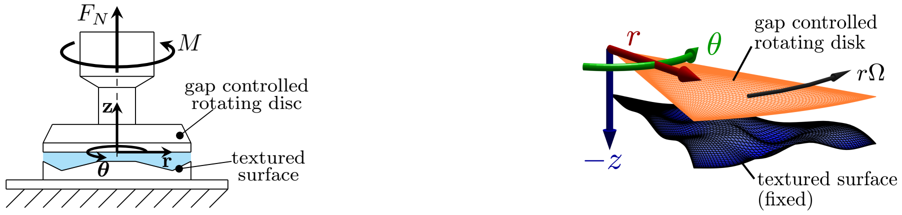
3. Introduction

Surface textures reduce friction in lubricated sliding contact [1]. In many practical engineering applications with lubricated sliding contact, reducing energy loss, wear, friction-induced noise, and improving compactness are important design objectives [2, 3]. Since the roughness of a sliding surface primarily affects friction performance, a comprehensive set of studies has been conducted to understand how changes in surface roughness can reduce friction. Many existing studies focus on creating an array of dimples with circular or other shapes. It has been demonstrated that circular micro dimples fabricated using abrasive jet machining and laser beam machining with different sizes can help reduce lubrication friction [4]. A theoretical model of micro-dimple hydrodynamics has been introduced, and has been verified experimentally showing that surface micro structures enhance frictional performance [1]. Previous studies have not been limited to only circular dimple shapes. More recent studies show that other specific surface texture top profile shapes can further improve frictional and load-carrying performance [5–7], while other studies also investigated the effect of depth profile changes as well [8, 9]. A series of recent studies tried to further enhance the frictional performance and load-carrying capacity by expanding the surface texture to free-form design of the full interface surface, and led to improvement of both objectives by an order of magnitude [2, 10]. These studies involved a reduced-dimension texture design parameterization as a strategy to improve computational efficiency for optimization studies, at the expense of reduced texture design resolution. The optimization studies were based on a Reynolds equation model, which is the simplest option for predicting frictional performance of lubricated sliding contacts. Several earlier numerical studies have been performed using the Reynolds equation to examine the effects of surface textures on friction reduction [11–13]. However, solving a fluid flow requires a computationally expensive solution of a nonlinear system of equations (even when based on a simplified governing equation, such as the Reynolds equation). Here a more efficient method for solving the texture optimization problem for lubricated sliding contact is proposed by linearizing the Reynolds equation, and sequentially updating the approximated linearization.

4. Problem Formulation

4.1. Problem Statement

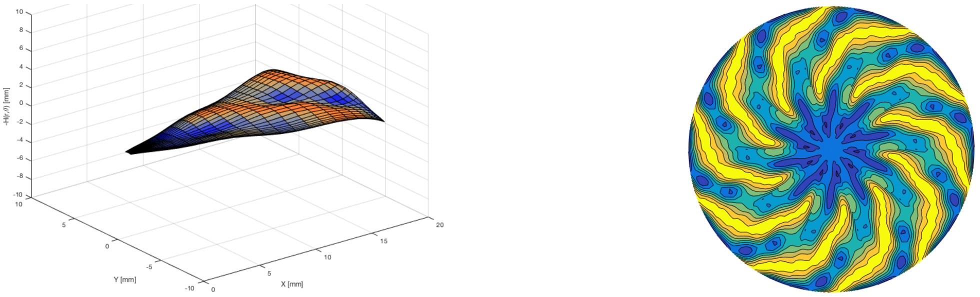
Using a rotational tribo-rheometer setup given in Fig. 1, we are designing a two-dimensional texture profile for the fixed bottom plate. The nominal gap height between the top and bottom plates is fixed, and the torque and normal force on the moving flat surface is measured [9]. The goal of this study is to decrease friction within the system while simultaneously increasing the normal force produced by the surface texture. Recent studies demonstrated that these two objectives could be achieved with asymmetric depth profile surface textures [2,9,14]. In this study, we optimize the height profile of the surface texture using a computationally efficient linearization technique applied to the Reynolds equation. The design variable (gap height) has the same dimensions as the computational mesh, i.e., we are designing the gap height at each mesh point. Figure 1a shows the front view of the experiment setup of this study. Two disks are separated by Newtonian fluid, with the flat disk rotating on the top, and the textured surface fixed at the bottom. To reduce the computing complexity, we divide the full disk into $N_{\mathbf{tex}}$ periodic sectors. In this study, the choice of $N_{\mathbf{tex}}$ will affect our results, and we chose $N_{\mathbf{tex}} = 10$. Figure 1b is a closer look of the design domain. Top plate is flat, rotating in the θ direction, while an arbitrary textured surface is fixed at the bottom. Eventually, the texture surface design will be mapped into a full disk by applying the periodic boundary condition.



(a) Front view (after [2]) **(b)** Simulated experiment setup with one sector (after [14])

Figure 1: The experiment setup adapted from previous studies [2,9]. (a) Front view of the setup, with one flat disc rotating on the top, and a fixed textured surface at the bottom. The two disks are separated by Newtonian fluid. (b) One sector of the experiment setup, with the axes showing the directions used in this study.

The surface texture profile when $F = 7.03$ and $\tau = 0.737$ from the study by Lee et al. [2] is used as the starting design point for all optimization studies presented here. This benchmark design is shown in Fig. 2. By applying different upper bounds on shear loads, we can obtain a set of optimal points. This procedure will be discussed in more detail in Sections 4.3, 4.4, and 6.



(a) Surface of one sector of the starting design **(b)** Full disk plot of the starting point

Figure 2: Benchmark design used as a starting design point for optimization studies in this paper. This benchmark design is the optimal surface texture profile obtained during a previous study [2]. The corresponding normal force is $F_N = 7.03$ N, and the apparent shear load is $\tau = 0.737$ (normalized). (a) Surface for a single disk sector of the benchmark design. (b) Full disk image of the benchmark design.

4.2. Numerical Model for Texture Surface Hydrodynamics

Here we assume that the operating lubricant can be modeled as an incompressible Newtonian fluid with constant viscosity, and that the flow occurs at low Reynolds numbers so that inertial terms can be neglected. We also assume that the gap height is small compared to other length scales in the problem so that gradients in the flow direction can be neglected. Based on these assumptions, the system may be modeled using the Reynolds equation [15] in cylindrical coordinates:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(rh^3\frac{\partial p}{\partial r}\right) + \frac{1}{r}\frac{\partial}{\partial \theta}\left(\frac{h^3}{r}\frac{\partial p}{\partial \theta}\right) = 6\eta\Omega\frac{\partial h}{\partial \theta}, \tag{1}$$

where p is the pressure, h is the gap height, η is the fluid viscosity, and Ω is the angular velocity. The Reynolds equation is a linear, second-order partial differential equation with a non-constant coefficient for pressure that satisfies both conservation of mass and momentum. The velocity boundary conditions used to derive Eqn. (1) are:

$$\text{at } z=0: \; v_\theta = r\Omega \text{ , and } v_r = v_z = 0 \tag{2a}$$

$$\text{at } z=h: \; v_\theta = v_r = v_z = 0. \tag{2b}$$

Integrating conservation of momentum and applying boundary conditions produces the following velocity field:

$$v_r = \frac{1}{2\eta}\frac{\partial p}{\partial r}\left(z^2 - zh\right) \tag{3a}$$

$$v_\theta = \frac{1}{2\eta}\frac{1}{r}\frac{\partial p}{\partial \theta}\left(z^2 - zh\right) + r\Omega\left(1 - \frac{z}{h}\right). \tag{3b}$$

Schuh et al. [14] developed MATLAB$^{\text{TM}}$ code for solving Eqn. (1) using the pseudo-spectral method, which transforms the partial differential equation into an approximate system of algebraic equations:

$$K\underline{p} = \underline{f}, \tag{4}$$

where:

$$K = \frac{-\varphi}{R_o - R_i}(I \otimes D)^T (M \otimes M)(I \otimes R) H^3 (I \otimes D) \frac{-(R_o - R_i)}{\varphi}(D \otimes I)^T (M \otimes M)\left(I \otimes R^{-1}\right) H^3 (D \otimes I) \tag{5a}$$

$$\underline{f} = 3\eta\Omega (R_o - R_i)(M \otimes M)(I \otimes R)(D \otimes I)\underline{h}. \tag{5b}$$

where φ is the total angle for a periodic sector, R_o is the outer radius, R_i is the inner radius, I is the identity matrix, D is a full matrix containing coefficients for approximating the required derivatives, M is a diagonal matrix of Gauss-Lobatto-Legendre (GLL) quadrature weights, R is a diagonal matrix of radius values from R_i to R_o, $\underline{h}$ is a vector, reshaped from the matrix containing the gap height values at each grid point, H is a matrix with diagonal elements from the vector h, and $\otimes$ is the Kronecker product. It was assumed in this derivation that the gap height is periodic in the θ direction.

A previous design optimization study used h as an input to Eqn. (4), which was then solved to obtain the pressure field, and the velocity field is computed using this pressure field [2]. These fields were then used to determine the objective functions for a given gap height profile (load capacity and a friction metric). In this previous strategy the Reynolds equation solver was treated as a *black-box* simulation. Each new design candidate tested by the optimization algorithm required simulation, contributing to significant computational expense as Reynolds equation structure was not leveraged for more efficient solution. If we examine Eqn. (4) carefully, however, observations reveal an opportunity to capitalize on problem structure utilizing both Simultaneous Analysis and Design (SAND) method [16] and linearization. SAND involves simultaneous Reynolds equation solution and gap height (texture design) determination. The first step is to rewrite Eqn. (4) in the following form:

$$\begin{bmatrix} K_r & K_\theta \end{bmatrix} \begin{bmatrix} \underline{h}^3 \odot \underline{p_r} \\ \underline{h}^3 \odot \underline{p_\theta} \end{bmatrix} = A\underline{h}, \tag{6}$$

where p_r is the gradient of the pressure in the r direction, p_θ is the gradient of pressure in the θ direction, $\odot$ is element-wise multiplication (Hadamard product), and:

$$K_r = \frac{-\varphi}{R_o - R_i}(I \otimes D)^T (M \otimes M)(I \otimes R) \tag{7a}$$

$$K_\theta = \frac{-(R_o - R_i)}{\varphi}(D \otimes I)^T (M \otimes M)\left(I \otimes R^{-1}\right) \tag{7b}$$

$$A = 3\eta\Omega (R_o - R_i)(M \otimes M)(I \otimes R)(D \otimes I). \tag{7c}$$

In Eqn. (6), the design variable h appears on both the left and right hand side of the equation, and the equation depends nonlinearly on the gap height. Using a SAND strategy, we combine our design and analysis variables into a new independent optimization variable, denoted $\underline{x}$ here:

$$\underline{x} = \left[\underline{h},\, \underline{p_r},\, \underline{p_\theta}\right]^{\mathsf{T}}. \tag{8}$$

To make Eqn. (6) linear in our design variable, we linearize $\underline{h}^3 \odot \underline{p_r}$ and $\underline{h}^3 \odot \underline{p_\theta}$ near their tangent planes using a multivariate Taylor series expansion:

$$\underline{h}^3 \odot \underline{p_r} = \underline{h_0}^3 \odot \underline{p_{r_0}} + 3\underline{h_0}^2 \odot \underline{p_{r_0}} \odot \left(\underline{h} - \underline{h_0}\right) + \underline{h_0}^3 \odot \left(\underline{p_r} - \underline{p_{r_0}}\right) \tag{9a}$$

$$\underline{h}^3 \odot \underline{p_\theta} = \underline{h_0}^3 \odot \underline{p_{\theta_0}} + 3\underline{h_0}^2 \odot \underline{p_{\theta_0}} \odot \left(\underline{h} - \underline{h_0}\right) + \underline{h_0}^3 \odot \left(\underline{p_\theta} - \underline{p_{\theta_0}}\right), \tag{9b}$$

where $\underline{h_0}$ is a nominal gap height, and $\underline{p_{r_0}}$ is the pressure gradient in the r direction and $\underline{p_{\theta_0}}$ is the pressure gradient in the θ direction for the nominal gap height. Substituting Eqns. (9a) and (9b) into Eqn. (6), using our new design variable $\underline{x}$ and simplifying we obtain:

$$\left(\begin{bmatrix} K_r & K_\theta \end{bmatrix} \begin{bmatrix} 3H_0^2 P_{r_0} & H_0^3 & \mathbf{0} \\ 3H_0^2 P_{\theta_0} & \mathbf{0} & H_0^3 \end{bmatrix} - \begin{bmatrix} A & \mathbf{0} & \mathbf{0} \end{bmatrix}\right) \underline{x} = \begin{bmatrix} K_r & K_\theta \end{bmatrix} \begin{bmatrix} 3\underline{h_0}^3 \odot \underline{p_{r_0}} \\ 3\underline{h_0}^3 \odot \underline{p_{\theta_0}} \end{bmatrix}, \tag{10}$$

where $H_0^2 P_{r_0}$ is a diagonal matrix with the elements $\underline{h_0}^2 \odot \underline{p_{r_0}}$, the diagonal matrix $H_0^2 P_{\theta_0}$ is composed of elements $\underline{h_0}^2 \odot \underline{p_{\theta_0}}$, H_0^3 is a diagonal matrix with the elements $\underline{h_0}^3$, and $\mathbf{0}$ is the zero matrix with appropriate dimensions. Equation (10) can be rewritten in the following compact form:

$$\hat{K}\underline{x} = \underline{\hat{f}}. \tag{11}$$

All of the design variables appear only on the left hand side of Eqn. (11), and the right hand size is known after the nominal gap height profile is given. Therefore, solving Eqn. (11) (with appropriate pressure boundary conditions [14]) produces the associated pressure field gradients for a given gap height design.

Our two objective functions are the normal force F_N and the non-dimensional shear stress τ^*. The normal force is obtained by integrating the pressure field. Solving Eqn. (11), however, produces pressure field gradients. The pressure field can be obtained from its gradient using:

$$\begin{bmatrix} I \otimes D \\ D \otimes I \end{bmatrix} \underline{p} = \begin{bmatrix} \underline{p_r} \\ \underline{p_\theta} \end{bmatrix}. \tag{12}$$

Once the pressure has been obtained, the normal force on the flat plate is calculated using

$$F_N \equiv \int_0^{2\pi} \int_{R_i}^{R_o} p r dr d\theta = N_{\mathbf{tex}} \int_{-\varphi/2}^{\varphi/2} \int_{R_i}^{R_o} p r dr d\theta, \tag{13}$$

where:

$$N_{\mathbf{tex}} = \frac{2\pi}{\varphi}. \tag{14}$$

This is calculated numerically using:

$$F_N = N_{\mathbf{tex}} \left(\frac{\varphi\left(R_o - R_i\right)}{4}\right) \left(w \otimes w\right)^T \left(I \otimes R\right) \underline{p}, \tag{15}$$

where w is the vector of GLL quadrature weights.

Pressure derivatives are used to determine the velocity using Eqns. (3a) and (3b), and derivatives of the velocity components are used to determine the shear stress on the top plate:

$$\tau_{z\theta}|_{z=0} = \eta \left(\frac{\partial v_\theta}{\partial z} + \frac{1}{r}\frac{\partial v_z}{\partial \theta}\right)|_{z=0}. \tag{16}$$

Substituting Eqn. (3b) and simplifying gives:

$$\tau_{z\theta}|_{z=0} = -\left(\frac{1}{2r}\frac{\partial p}{\partial \theta} h + \eta \frac{r\Omega}{h}\right). \tag{17}$$

The torque on the flat plate is calculated from the shear stress as:

$$M \equiv \int_0^{2\pi} \int_{R_i}^{R_o} \left(\tau_{z\theta}|_{z=0}\right) r^2 dr d\theta = N_{\mathbf{tex}} \int_{-\varphi/2}^{\varphi/2} \int_{R_i}^{R_o} \left(\tau_{z\theta}|_{z=0}\right) r^2 dr d\theta, \tag{18}$$

which is calculated numerically as:

$$M = N_{tex}\left(\frac{\varphi\left(R_o - R_i\right)}{4}\right)\left(w \otimes w\right)^T \left(I \otimes R^2\right)\underline{\tau_{z\theta}}. \tag{19}$$

The torque can then be used to determine the non-dimensional shear stress (objective function):

$$\tau^* \equiv \frac{\frac{2}{\pi R_0^3}M}{\frac{\eta R_0 \Omega}{h_0}} = \frac{2h_0}{\eta \pi R_0^4}\frac{M}{\Omega}. \tag{20}$$

Using the variables defined in x, we can rewrite Eqn. (17) as:

$$\tau_{z\theta}|_{z=0} = -\left(\frac{1}{\varphi r}p_\theta h + \eta\frac{r\Omega}{h}\right), \tag{21}$$

where we have used

$$\frac{\partial p}{\partial \theta} = \frac{2}{\varphi}p_\theta. \tag{22}$$

Equation (21) is non-linear in h and p_θ; since we previously linearized our problem about h_0 and p_{θ_0}, we must also linearize $\tau_{z\theta}$ about h_0 and p_{θ_0} for our analysis to be consistent. Performing Taylor series expansions for the non-linear terms produces:

$$p_\theta h = p_{\theta_0}h_0 + p_{\theta_0}\left(h - h_0\right) + \left(p_\theta - p_{\theta_0}\right)h = -p_{\theta_0}h_0 + p_{\theta_0}h + p_\theta h_0, \tag{23a}$$

$$\frac{1}{h} = \frac{1}{h_0} - \frac{1}{h_0^2}\left(h - h_0\right) = \frac{2}{h_0} - \frac{1}{h_0^2}h, \tag{23b}$$

which when substituted back into Eqn. (21) yields:

$$\tau_{z\theta}|_{z=0} = -\left(\left(\frac{p_{\theta_0}}{r\varphi} - \eta\frac{r\Omega}{h_0^2}\right)h + \frac{h_0}{r\varphi}p_\theta - \frac{p_{\theta_0}h_0}{r\varphi} + 2\eta\frac{r\Omega}{h_0}\right). \tag{24}$$

This is calculated numerically as:

$$\underline{\tau_{z\theta}} = -\begin{bmatrix} B_h & \mathbf{0} & B_{p_\theta}\end{bmatrix}\underline{x} - \underline{c}, \tag{25}$$

where

$$B_h = \frac{1}{\varphi}P_{\theta_0}\left(I \otimes R^{-1}\right) - \eta\Omega\left(I \otimes R\right)\left(H_0^2\right)^{-1} \tag{26a}$$

$$B_{p_\theta} = \frac{1}{\varphi}H_0\left(I \otimes R^{-1}\right) \tag{26b}$$

$$\underline{c} = -\frac{1}{\varphi}P_{\theta_0}\left(I \otimes R^{-1}\right)\underline{h_0} + 2\eta\Omega\left(I \otimes R\right)\underline{h_0}^{-1}, \tag{26c}$$

and P_{θ_0} is a diagonal matrix of elements $\underline{p_{\theta_0}}$, H_0^2 is a diagonal matrix of elements $\underline{h_0}^2$, and $\underline{h_0}^{-1}$ signifies an element-wise inversion of $\underline{h_0}$. Equation (25) can be used with Eqns. (19) and (20) to calculate the non-dimensional shear stress.

4.3. Multi-objective Optimization

Since we are solving a multi-objective problem with two competing objective functions, the solution sought is a Pareto frontier [17] that expresses the tradeoff between these two objectives. The ε-constraint method [18] is a commonly used technique to convert a multi-objective problem into a set of single-objective optimization problems. This is done by optimizing a single objective function while using the other objectives functions as inequality constraints with bounds. By adjusting these bounds for the objective functions posed as inequality constraints, we generate a set of problems. The solution of each of these problems produces a single point on the Pareto frontier.

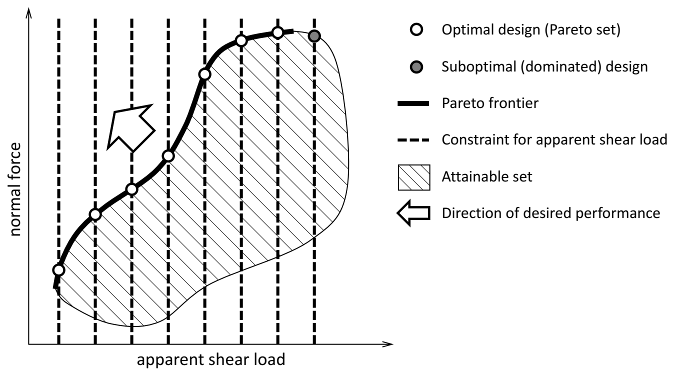

normal force
apparent shear load
Optimal design (Pareto set)
Suboptimal (dominated) design
Pareto frontier
Constraint for apparent shear load
Attainable set
Direction of desired performance

Figure 3: Schematic showing how the ε-constraint method is used for our problem. We optimize one objective function (here, maximizing normal force) while constraining the other metric using an inequality constraint (here, constraining apparent shear load to be less than a certain value). The incrementally varied maximum bound on shear load is shown as a dashed line. Open circles are the optimal design points obtained from each of the optimization subproblems. The gray circle is an example of a dominated (suboptimal) design point. These optimal designs are on the Pareto frontier, shown with a thick solid black line.

Figure 3 demonstrates the conflict between our two objectives, 1) minimizing the shear load, which is represented by normalized apparent shear viscosity, expressed as η_a/η_0, and 2) maximizing the normal force. Both these objectives help reduce friction in this lubrication problem, which is the main source of energy loss. From the Pareto frontier we can tell that, as normal force load F_N increases, the apparent shear stress increases inevitably. The feasible region for our design objectives is shaded; all optimal designs lie within this region. We optimized the normal force with respect to a given apparent shear load, which was used as our inequality constraint. The dashed black lines are different apparent shear load values, and the white circles are the optimal normal force values obtained for each given shear load bound. The open circles lie on the boundary of the attainable set. Connecting the white circles gives our Pareto front of optimal solutions, which is shown with the solid black line. The gray circle at the top right is an example of a *dominated point*; it is dominated because feasible solutions exist where both objectives can be improved simultaneously, as evident in Fig. 3. The texture designs that give this Pareto front are the optimal texture designs.

4.4. Trust Region and Step Size

An adaptive trust region method is essential to control linearization error, and helps improve solution convergence [19]. Here we implement a trust region method that involves five steps (described below).

4.4.1. Initialize Parameters

- Specify initial starting point ($\underline{x_0}$) and initial trust region radius (Δ_0). For gap height h, the initial trust region radius is set to be 10% of the gap height value at each grid point, and for the gradient of pressure values p_r and p_θ, the initial trust region radius is set to be 100.
- Set the trust region constants to: $\eta_1 = 0.01$, $\eta_2 = 0.9$, $\gamma_1 = \gamma_2 = 0.5$.

In the following development, we use $f(x_k)$ to indicate the objective function based on the original nonlinear equations, and $m(x_k)$ to represent the approximation of the objective function.

4.4.2. Step Size Calculation

Based on the linearized problem, a linear program with a trust region constraint is solved to determine a step s_k to determine a new design $x_k + s_k$ (trial point) that is inside the trust region.

4.4.3. Acceptance of Trial Point

Define the acceptance criteria to be:

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}. \tag{27}$$

If $\rho_k \geq \eta_1$, then set the new design to $x_{k+1} = x_k + s_k$. If this is not the case (i.e., $\rho_k < \eta_1$), the linear approximation is not sufficiently accurate, and we stay at the same point: $x_{k+1} = x_k$.

4.4.4. Trust Region Radius Update

We update the trust region depending on how the trial point $(x_k + s_k)$ performs:

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \rho_k \geq \eta_2; \\ [\gamma_2 \Delta_k, \Delta_k] & \rho_k \in [\eta_1, \eta_2]; \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \rho_k < \eta_1. \end{cases} \tag{28}$$

By comparing ρ_k with η_1 and η_2, the trust region can be adjusted. If the linear approximation is very accurate, (i.e. the acceptance ρ_k is larger than η_2), the trust region will be expanded. On the contrary, if ρ_k is small, the trust region will be shrunk accordingly. In the implementation here, constant values were used within the specified ranges for each of the three cases.

4.4.5. Step Size

Define $s_k = x_{k+1} - x_k$, and α to be the step length we move along the s direction. If $f(x_{k+1}) < f(x_k)$, $x_{k+1} = x_k + \alpha * s_k$. In this implementation, we used a fixed value for α of 0.01.

With this method, the trust region size is adjusted adaptively to maintain acceptable linearization accuracy. When the algorithm uses a good search direction, and the estimate is accurate, the trust region is expanded. If the LP algorithm does not converge, produces a poor search direction, or does not estimate function values accurately, then we shrink the trust region to compensate. Inspired by the move limit method used in truss design method developed by John et al. [20], we choose a new design point with α as the move limit if no improvement is observed. Instead of adaptively updating α as the paper suggested, we use a constant value, 0.01, in this study. Studying strategies for updating alpha adaptively is a topic left for future work.

5. Verification

We compare our results to those reported previously for the same problem, but using a non-linear programming (NLP) solution method with a spline texture design representation [10]. Here we use previous optimal designs from the NLP solution as starting points for the sequential linear programming (SLP) strategy introduced here.

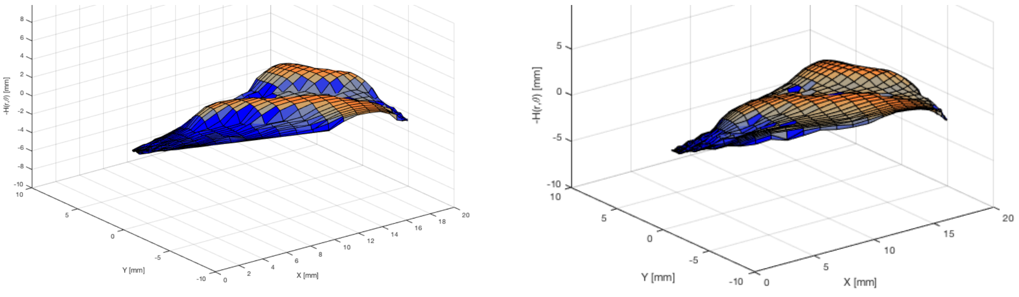
Table 1: Comparison of SLP to the full non-linear optimization method where the result from the non-linear optimization are used as starting points for the SLP solution method. The SLP results match the non-linear optimization results exactly, validating the SLP formulation.

Shear Viscosity	0.76	0.73	0.71	0.69	0.63	0.57	0.50
Normal Force (Non-linear)	6.25	6.00	5.75	5.50	5.00	4.25	3.25
Normal Force (SLP)	6.25	6.00	5.75	5.50	5.00	4.25	3.25
Error [%]	0	0	0	0	0	0	0

The results of the SLP and NLP solution strategies are compared in Table 1. SLP obtains the same optimal normal force values as NLP, validating our method. However, the linear program cannot produce an improvement over the NLP results as the linear programming method is only able to identify locally optimal points.

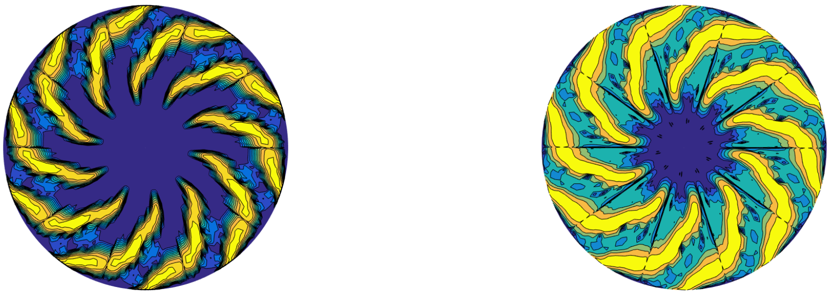
6. Results and Discussion

Using the results from the previous NLP study as our starting points (where $F_N = 7.37$ N), we generated the Pareto set by using different values for the shear load bound (as shown in Section 4.2). Sample surface texture designs from the Pareto set are shown in Figs. 4a and 4b. (The reader can generate these plots using the code provided for numerical experiment replication [21]). The Pareto set was obtained using the optimizer MOSEK$^{\text{TM}}$ with a single start method coupled with our trust region strategy. This improves the ability to find optimal points that may be far away from the initial design point.



(**a**) Sample surfaces, $\tau^* = 0.481$, $F = 2.91$ (**b**) Sample surfaces, $\tau^* = 0.632$, $F = 5.20$

Figure 4: Two sample surfaces for a single disk sector. Each surface texture design corresponds to a different point on the Pareto frontier, as shown in Fig. 6.



(**a**) Sample full disk plot, $\tau^* = 0.481$, $F = 2.91$ (**b**) Sample full disk plot, $\tau^* = 0.632$, $F = 5.20$

Figure 5: Two sample texture designs for a full disk. Each disk image corresponds to a different point on the Pareto frontier, as shown in Fig. 6.

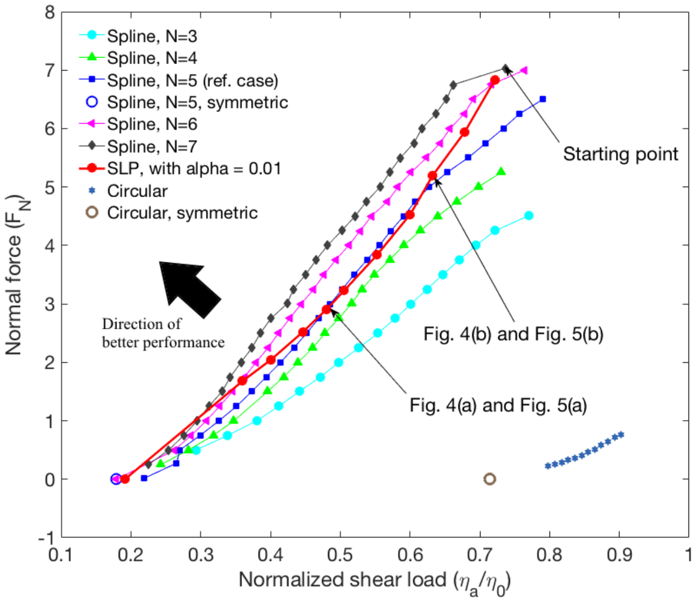


Figure 6: Comparison of the Pareto sets obtained using the NLP method [10] and the SLP method introduced here. SLP produces better results than NLP when compared to spline surface representations of order $N = 3$, 4, and 5. The NLP solution with spline order $N = 5$ was chosen as the reference case. The SLP solution performs similar to the reference case in the region $0.469 < \tau_* < 0.628$, but performs better than the reference case at both high and low τ^*. The NLP solutions with spline order $N = 6$ and 7 (improved resolution) perform better than the SLP solution.

Figure 6 compares the Pareto fronts obtained using the SLP solution to those obtained using NLP with varying spline representation orders ($N \in \{3,4,5,6,7\}$), where $N = 5$ is used as a reference solution. The SLP solution performs similar to the reference NLP solution in the region $0.469 < \tau^* < 0.628$, but performs better than the reference case at both high and low τ^* values. In the region where $0.469 < \tau^* < 0.628$, there are points where the SLP method performs worse than the NLP method (maximum 25% deviation in normal force value between the two methods at $\tau^* = 0.5$). This may be due to the choice of trust region and step size. Another factor may be the existence of multiple distinct surface texture designs with the same normal force value for a give shear load, as shown in Fig. 2.

The greatest advantage of using the SLP strategy is realized when comparing computational expense for solution. Table 2 compares the total computation time required (on average) to find an optimal point when using different N values for both the NLP and SLP methods. The reported computation time was obtained using a dual-core Intel Core$^{\text{TM}}$ i5-4250U processor for all studies. As with many numerical optimization problems, we see that there is a trade-off here between solution quality and computational expense. Introducing the SLP strategy allows us to reduce computational expense significantly while achieving solution quality comparable to many of the NLP results. Table 3 shows how much faster the SLP solution is compared to the different NLP solutions. Even for the lowest-order spline design representation, the NLP method is almost twice as slow as SLP. When comparing the highest-order NLP solution time to SLP, we find that SLP is almost 20 times faster.

Table 2: Computational Time to Find An Optimal Point (on Average)

$N = 3$ (NLP)	$N = 4$ (NLP)	$N = 5$ (NLP)	$N = 6$ (NLP)	$N = 7$ (NLP)	SLP
3.7 min	7.3 min	12.2 min	17.5 min	25.1 min	1.3 min

Table 3: Relative Computational Time Comparison

NLP with reduced design representation	$N = 3$	$N = 4$	$N = 5$	$N = 6$	$N = 7$
Time improvement using SLP [%]	185	462	838	1,246	1,831

7. Conclusion

Several numerical studies have been performed based on a Reynolds equation model to examine the effects of surface texture design variation on friction reduction and normal force generation. Previous optimization studies achieved efficient solution by reducing design representation dimension using spline texture representation, and solving the optimization problem using nonlinear programing (NLP). In these previous studies the simulation was treated as a black box. Here, model structure was leveraged to formulate a sequential linear programming (SLP) strategy with the objective of simultaneously improving design representation fidelity and computational efficiency. Improved design fidelity may support identification of new types of designs and physical mechanisms to further improve system performance over the previous NLP results, although this consideration is a topic of ongoing work and not within the scope of this article.

Numerical studies presented here indicate significant improvements in computational efficiency, while maintaining reasonable solution quality. While NLP solutions using high-order spline representations produced higher-performance designs than SLP, the SLP solution was at least an order of magnitude faster. Another improvement is robustness to starting point. The previous NLP strategy required a multi-start approach to converge to a good solution reliably. In this study it was shown that a single well-selected starting point was sufficient for generating all Pareto-optimal points. This method was demonstrated using a sector of the sliding surface, but it could be extended to other models based on a semi-circle, or even the full rotating disk.

The SLP method, however, has several clear limitations. As SLP cannot explore outside the trust region, once the trial point drops inside the feasible region it can only find a local optimum. Thus, it cannot improve the solution beyond the best results from the previous NLP studies. In addition, because of linearization error, the SLP method produces a design with reduced performance when τ^* values are between 0.469 and 0.628.

Another limitation of the current implementation relevant to moving trust regions should be highlighted. When using the ε-constraint method, a multi-objective optimization problem is decomposed into a set of single-objective subproblems. A typical strategy in the solution of these subproblems is to use the result of one subproblem as a starting point for a neighboring subproblem. This typically helps reduce solution expense. In this particular implementation, however, if we use the solution of a subproblem with a larger shear load bound as a starting point for a subproblem with a smaller shear load bound, the initial point will violate the shear load constraint. When the linear program (LP) solver detects a constraint violation, it attempts to satisfy the constraint by moving the solution far from the current design. A large single step is taken, reducing iterations. Conversely, when solving a subproblem using a starting point obtained from a lower shear load subproblem, the trust region limits progress. Even if the trust region radius is adapted, effective design exploration is hampered. To address this issue, here we began the process by solving the largest shear load subproblem first, and decremented the shear load constraint. Thus, in this study, we solve this problem beginning with the largest shear load case and using the result of this subproblem, lower shear load subproblems are solved. An improvement of this trust region updating issue is a topic of ongoing work.

Several SLP method improvements and research questions have been identified for future work. A more sophisticated step size update strategy may help accelerate convergence, and improve exploration of a wider range of designs. SLP method robustness should be improved, including the ability to start from arbitrary design points (such as a flat plane) and converge to high-performance solutions. Finally, alternative formulations may better capture the underlying physical mechanisms that lead to improved performance. For example, a quadratic approximation may be a better approximation for the governing equations and objective functions. While this may help drive the solution method to better designs, an LP can no longer be formulated. A problem with quadratic constraints will not be as easy to solve as an LP, but may still support use of problem structure for solution efficiency better than previous general NLP studies.

8. References

[1] I. Etsion, Improving tribological performance of mechanical components by laser surface texturing, *Tribology Letters*, vol. 17, no. 4, pp. 733–737, 2004.

[2] Y. H. Lee, J. K. Schuh, R. H. Ewoldt, and J. T. Allison, Enhancing full-film lubrication performance via arbitrary surface texture design, *Journal of Mechanical Design, in press*, 2017.

[3] C. Dong, C. Yuan, L. Wang, W. Liu, X. Bai, and X. Yan, Tribological properties of water-lubricated rubber materials after modification by mos2 nanoparticles, *Scientific Reports*, vol. 6, p. 35023 (12pp), 2016.

[4] M. Wakuda, Y. Yamauchi, S. Kanzaki, and Y. Yasuda, Effect of surface texturing on friction reduction between ceramic and steel materials under lubricated sliding contact, *Wear*, vol. 254, no. 3–4, pp. 356–363, 2003.

[5] H. Yu, X. Wang, and F. Zhou, Geometric shape effects of surface texture on the generation of hydrodynamic pressure between conformal contacting surfaces, *Tribology Letters*, vol. 37, no. 2, pp. 123–130, 2010.

[6] C. Shen and M. M. Khonsari, Effect of dimples internal structure on hydrodynamic lubrication, *Tribology Letters*, vol. 52, no. 3, pp. 415–430, 2013.

[7] M. Fesanghary and M. M. Khonsari, On the optimum groove shapes for load-carrying capacity enhancement in parallel flat surface bearings: theory and experiment, *Tribology International*, vol. 67, pp. 254–262, 2013.

[8] S. M. Hsu, Y. Jing, D. Hua, and H. Zhang, Friction reduction using discrete surface textures: principle and design, *Journal of Physics D: Applied Physics*, vol. 47, no. 33, p. 335307 (12pp), 2014.

[9] J. K. Schuh and R. H. Ewoldt, Asymmetric surface textures decrease friction with Newtonian fluids in full film lubricated sliding contact, *Tribology International*, vol. 97, pp. 490–498, 2016.

[10] Y. H. Lee, J. K. Schuh, R. H. Ewoldt, and J. T. Allison, Shape parameterization comparison for full-film lubrication texture design, in *ASME 2016 IDETC/CIE, Volume 2B: 42nd Design Automation Conference*, (Boston, MA), p. V02BT03A037 (11pp), ASME, August 2016.

[11] A. Ronen, I. Etsion, and Y. Kligerman, Friction-reducing surface texturing in reciprocating automotive componenets, *Tribology Transactions*, vol. 44, pp. 359–366, 2001.

[12] R. Siripuram and L. Stephens, Effect of deterministic asperity geometry on hydrodynamic lubrication, *Journal of Tribology*, vol. 126, pp. 527–534, 2004.

[13] Y. Qiu and M. M. Khonsari, On the prediction of cavitation in dimples using a mass-conservative algorithm, *Journal of Tribology*, vol. 131, pp. 041702–1–11, 2009.

[14] J. K. Schuh, Y. H. Lee, J. T. Allison, and R. H. Ewoldt, Design-driven modeling of surface-textured full-film lubricated sliding: validation and rationale of nonstandard thrust observations, *Tribology Letters*, vol. 65, no. 2, p. 35 (17pp), 2017.

[15] O. Reynolds, On the theory of lubrication and its application to Mr. Beauchamp Tower's experiments, including an experimental determination of the viscosity of olive oil, *Proceedings of the Royal Society of London*, vol. 40, no. 242-245, pp. 191–203, 1886.

[16] E. J. Cramer, J. J E Dennis, P. D. Frank, R. M. Lewis, and G. R. Shubin, Problem formulation for multidisciplinary optimization, *SIAM Journal on Optimization*, vol. 4, no. 4, pp. 754–776, 1994.

[17] M. J. Reddy and D. N. Kumar, Elitist-Mutated multi-objective particle swarm optimization for engineering design, in *Encyclopedia of Information Science and Technology, Third Edition* (M. Khosrow-Pour, ed.), pp. 3534–3545, Hershey, PA, USA: IGI Global, 2015.

[18] Y. V. Haimes, L. S. Lasdon, and D. A. Wismer, On a bicriterion formation of the problems of integrated system identification and system optimization, *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-1, pp. 296–297, 7 1971.

[19] P. Papalambros and D. Wilde, *Principles of Optimal Design: Modeling and Computation*. Cambridge University Press, 2000.

[20] K. V. John, C. V. Ramakrishnan, and K. G. Sharma, Minimum weight design of trusses using improved move limit method of sequential linear programming, *Computers & Structures*, vol. 27, no. 5, pp. 583–591, 1987.

[21] C. Lin, K. J. Schuh, and Y. H. Lee, Efficient optimal surface texture design using linearization. `https://github.com/yonghoonlee/SLP_TextureDesign_ReyEQ`, 2017.