Dynamic Kelvinlets:
Secondary Motions based on Fundamental Solutions of Elastodynamics

Pixar Technical Memo #18-05

FERNANDO DE GOES, Pixar Animation Studios
DOUG L. JAMES, Pixar Animation Studios and Stanford University

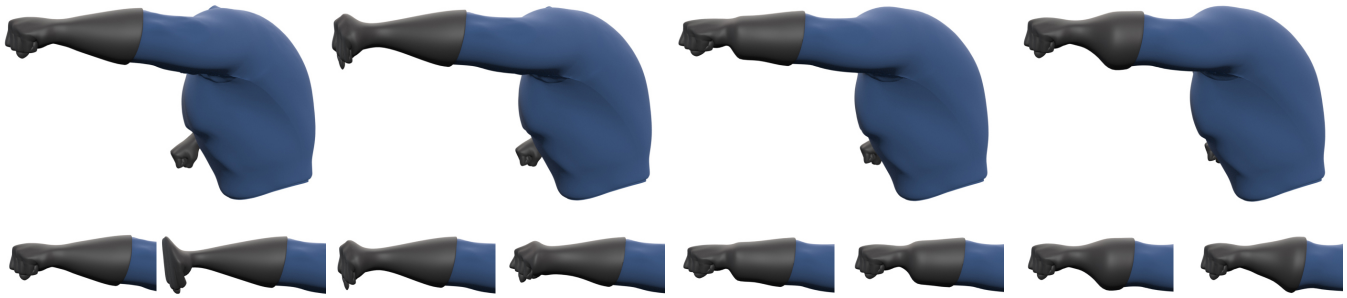


Fig. 1. **Dynamic Kelvinlets:** A punch animation is augmented with impact waves generated by dynamic Kelvinlets, a novel approach for computing volumetric deformations with real-time feedback based on analytical solutions of linear elastodynamics. Top row shows elastic waves propagated across the character's fist and forearm after impact with an imaginary wall. Bottom row shows a close-up of the animation sequence. ©Disney/Pixar

We introduce *Dynamic Kelvinlets*, a new analytical technique for real-time physically based animation of virtual elastic materials. Our formulation is based on the dynamic response to time-varying force distributions applied to an infinite elastic medium. The resulting displacements provide the plausibility of volumetric elasticity, the dynamics of compressive and shear waves, and the interactivity of closed-form expressions. Our approach builds upon the work of de Goes and James [2017] by presenting an extension of the regularized Kelvinlet solutions from elastostatics to the elastodynamic regime. To finely control our elastic deformations, we also describe the construction of compound solutions that resolve pointwise and keyframe constraints. We demonstrate the versatility and efficiency of our method with a series of examples in a production grade implementation.

CCS Concepts: • **Computing methodologies** → **Physical simulation**; **Procedural animation**;

Additional Key Words and Phrases: elastic waves, linear elastodynamics.

ACM Reference Format:
Fernando de Goes and Doug L. James. 2018. Dynamic Kelvinlets: Secondary Motions based on Fundamental Solutions of Elastodynamics Pixar Technical Memo #18-05 . *ACM Trans. Graph.* 37, 4, Article 81 (August 2018), 10 pages. https://doi.org/10.1145/3197517.3201280

1 INTRODUCTION

Physically based simulation is a mainstay of computer graphics widely used to animate virtual characters and natural phenomena. Despite their success, existing physics-driven techniques still have several downsides that interrupt the animation process. For instance, the dynamics computations are cumbersome due to numerical solves and time-stepping conditions. Another major impediment is the tedious setup phase which often involves volumetric meshing. These limitations motivate digital artists to use simpler models that capture the visual cues at interactive rates, especially for small-scale and secondary deformations where approximated solutions may suffice.

In this paper, we introduce a new interactive tool for the generation of physically based dynamics with elastic deformations. Our approach is based on the extension of the elastostatic regularized Kelvinlets [de Goes and James 2017] to dynamics. This is achieved by deriving novel fundamental solutions of elastodynamics for spatially regularized and time-varying forces applied to an infinite continuum. The resulting *dynamic Kelvinlets* lead to analytical closed-form expressions that define wave-like volumetric displacements. Consequently, we can animate deformable models free of any geometric discretization, computationally intensive solve, or stability restriction. Instead, our elastic deformations can be evaluated rapidly both in space and time, with interactive control of wave speed and volume compression. While not intended for general simulation tasks, dynamic Kelvinlets are well-suited to procedural visual effects such as jiggling, ripples, denting, and blasts (Figure 1).

Our contributions encompass several types of dynamic Kelvinlets. The first and most fundamental solution is the *impulse dynamic Kelvinlet*, which describes volumetric waves caused by an impulse load. Based on this solution, we derive the elastic response to steady push-like forces, which we call the *push dynamic Kelvinlet*. The latter produces permanent deformations that converge in the quasi-static regime to the regularized Kelvinlets [de Goes and James 2017]. We also present time-varying deformations generated by affine loads such as twist, scale, and pinch, and the construction of constrained and compounded elastic waves.

Authors' addresses: Fernando de Goes, Pixar Animation Studios, fernando@pixar.com; Doug L. James, Pixar Animation Studios, Stanford University, djames@pixar.com.

2 RELATED WORK

The dynamics of deformable models have been studied extensively in computer graphics, see survey [Nealen et al. 2006]. Since simulation techniques are laborious, many practitioners prefer simpler approaches that are easy to setup and interactive. This has motivated several approximations of physical models especially for small-scale dynamics. Examples include ad hoc simulation of ropes and springs [Barzel 1997], contact modeling [Pauly et al. 2004], shape matching [Müller et al. 2005], and oriented particles [Müller and Chentanez 2011], to cite a few. Our work exploits linear elastodynamics to generate setup-free and interactive animations with elastic wave-like deformations.

Mass-spring systems are widely used in computer animation for jiggling motions and can be found in many commercial packages, e.g., [Autodesk 2016; Side Effects 2018]. Unfortunately, these solvers are subject to time-stepping restrictions that depend on mesh resolution and model stiffness. Recent efforts have improved the robustness to large time-steps at the cost of degrading numerical convergence [Bouaziz et al. 2014; Liu et al. 2013]. In contrast, we provide analytical elastodynamic solutions that can be evaluated for any point and at any time. We thus circumvent time-stepping restrictions, geometric discretization, or numerical solves.

The work of von Funck et al. [2007] employed mass-spring sets to steer volume-preserving deformations that create secondary dynamics by advecting points along user-crafted vector fields. Similarly, Angelidis and Singh [2007] used the kinematics of a character skeleton to trigger analytical swirling displacements. Our approach shares similar properties with these techniques, in particular, it guarantees foldover-free deformations guided by time-varying displacements. Instead of using motion controllers, we rely on analytical solutions to the elastic wave equation to produce physically based animations. Moreover, we address a wider family of deformations that incorporates volumetric responses to elastic materials with compression. Consequently, our method can capture the propagation of elastic waves caused by shearing and volume compression.

Surface water waves are commonly animated using height-field procedural methods. They range from simple ripple effects based on a scalar wave equation [Kass and Miller 1990] to spectral methods for ocean waves [Tessendorf 2001], Lagrangian wave particles [Yuksel et al. 2007], wavefront tracking [Jeschke and Wojtan 2015], dispersive kernels [Canabal et al. 2016], and water wave packets [Jeschke and Wojtan 2017]. We instead address elastic waves that are not height-field deformations but rather volumetric fields. Our solutions also account for dipole-driven and shear waves, which are not present in fluid scenarios.

Model reduction techniques, such as linear modal analysis, can accelerate elastic simulations based on precomputed deformation modes [Pentland and Williams 1989]. The work of James and Pai [2002], for instance, used linear modal analysis augmented with rigid motion transfer functions in order to produce elastodynamic response textures for character animations. However, these methods require significant object-specific preprocessing work and exhibit linearization artifacts. Improved model reduction techniques can alleviate linearization artifacts by accounting for material nonlinearities [An et al. 2008], or by defining better motion subspaces such as those spanned by animation rigs [Hahn et al. 2012]. One remaining issue is that low-rank approximations to bulk deformation are still unable to reproduce elastic wave solutions.

Oscillatory motions can be authored using wiggly splines [Kass and Anderson 2008], or by replicating and time-shifting a root animation to subsequent rig elements [Shen et al. 2015]. The work of Schulz et al. [2014] combined wiggly splines with modal analysis for animating deformable objects with spacetime constraints. Our method also supports pointwise and keyframe constraints, but bypasses any expensive precomputation or memory overhead.

Our work introduces a dynamics version of the regularized Kelvinlets [de Goes and James 2017]. In the incompressible regime, it corresponds to a temporal extension of the regularized Stokeslets [Cortez 2001]. Our formulation provides closed-form expressions for the elastodynamic response to regularized loads in an unbounded space. This removes the singularities to the classical elastodynamic solution due to point impulses, which is often referred to as the Stokes's problem [Stokes 1849] and serves as the basis for quantitative techniques in seismology [Kausel 2006]. Seismic inversion methods, for example, leverage these singular solutions to detect and estimate the magnitude, location, and moments of seismic events from measured elastic waves [Aki and Richards 1980]. In contrast to seismology where singular sources are used to characterize far-field effects, dynamic Kelvinlets are suitable for deformations near the source centers such as secondary-wave animations around impact points.

Finally, we point out that the singular elastodynamic solutions are the foundation for boundary element methods (BEM) [Dominguez 1993]. However, these techniques depend on the discretization of 3D solid shapes and require costly dense linear solves per time-step. We instead investigate the use of regularized free-space deformations as a new artistic tool for animating elastic effects.

3 BACKGROUND

We begin by reviewing key concepts of linear elastodynamics upon which our formulation is based. Here we consider time-varying deformations of an infinite 3D medium formed by an isotropic and homogenous elastic material. For a thorough introduction to linear elastodynamics, we point the reader to [Kausel 2006].

Elastodynamics: In linear elasticity, the dynamics of an infinite 3D continuum is determined by the time-varying displacement field $\boldsymbol{u}:\mathbb{R}^3\times\mathbb{R}\rightarrow\mathbb{R}^3$ corresponding to the solution of:

$$m\,\partial_{tt}\boldsymbol{u} = \mu\Delta\boldsymbol{u} + \frac{\mu}{(1-2\nu)}\nabla(\nabla\cdot\boldsymbol{u}) + \boldsymbol{b}, \tag{1}$$

where $\boldsymbol{b}$ is a time-varying external body force, m is the mass density, μ is the the elastic shear modulus indicating the material stiffness, and ν is the Poisson ratio that controls the material compressibility. The mass density is a constant that scales both μ and $\boldsymbol{b}$. We can thus set $m=1$ without loss of generality. The equation of motion in (1) is known as the *elastic wave equation*, since it resembles a 3D wave equation with an additional divergence term that penalizes infinitesimal volume changes. Note that the displacements are defined over the infinite space and therefore no boundary conditions are needed, except for the Sommerfeld radiation condition which requires that the forces act as sources of elastic waves that radiate outward to infinity (see, e.g., [Aki and Richards 1980]).

Elastic Waves: We can analyze the displacement field $\boldsymbol{u}$ and the body force $\boldsymbol{b}$ using the scalar and vector potentials associated with their Helmholtz decomposition, i.e., $\boldsymbol{u} = \nabla\phi + \nabla\times\Phi$ and $\boldsymbol{b} = \nabla\psi + \nabla\times\Psi$. By substituting these potentials into (1), the elastic wave equation is decoupled into the following wave equations:

$$\begin{cases} \partial_{tt}\phi = \alpha^2\Delta\phi + \psi, & \text{(2a)} \\ \partial_{tt}\Phi = \beta^2\Delta\Phi + \Psi, & \text{(2b)} \end{cases}$$

with constant wave speeds $\beta = \sqrt{\mu}$ and $\alpha = \beta\sqrt{1 + 1/(1-2\nu)}$, and subject to Sommerfeld radiation conditions for out-going waves. Both potentials ϕ and Φ are now described by inhomogeneous wave equations, but with different wave speeds set to α and β respectively. Therefore, elastic waves can be seen as the result of two kinds of waves. First, we have the pressure waves (or *P-waves*) associated with the solution of (2a), which determines the volume oscillation in space and time generated by $\nabla\phi$ in response to $\nabla\psi$. Note that the resulting volume changes are contingent on the Poisson ratio ν used by the P-wave speed α. In addition, we have the shear waves (or *S-waves*) that produce divergence-free displacements $\nabla\times\Phi$ via (2b) in response to $\nabla\times\Psi$, with speed β dependent solely on the elastic shear modulus μ. Figure 3 illustrates the deformations caused by P- and S-waves separately in response to a localized translational body force. We also point out that P-waves are always faster than S-waves for any elastic material $\nu \in [0, 1/2)$, and these waves can only coincide when setting the unphysical value of $\nu = -\infty$.

Incompressibility: In the limit of $\nu = 1/2$, the divergence term in (1) becomes a hard constraint $\nabla\cdot\boldsymbol{u} = 0$ that ensures incompressible displacements. The corresponding elastic wave equation is then:

$$\partial_{tt}\boldsymbol{u} = \mu\Delta\boldsymbol{u} + \boldsymbol{b} - \nabla p \quad \text{subject to} \quad \nabla\cdot\boldsymbol{u} = 0, \tag{3}$$

where p is the pressure scalar field acting as a time-varying Lagrangian multiplier that enforces the divergence-free constraint. In this case, the scalar potential ϕ reduces to a trivial solution since $\nabla\cdot\boldsymbol{u} = \Delta\phi = 0$. One can further verify that the pressure field p cancels the scalar potential ψ of the body forces $\boldsymbol{b}$, i.e., $\nabla p = \nabla\psi$. As a consequence, the contribution of ϕ to the displacements $\boldsymbol{u}$ is zero, and only the S-waves are observable in the resulting dynamics.

Fundamental Solution: When the body force is a concentrated load due to a force vector $\boldsymbol{f}$ applied to a point $\boldsymbol{c}$ and at time zero, i.e., $\boldsymbol{b}(\boldsymbol{x}, t) = \boldsymbol{f}\,\delta(\boldsymbol{x}-\boldsymbol{c})\,\delta(t)$, the solution of (1) defines the fundamental solution of linear elastodynamics, which can be written as a linear combination of Dirac δ and Heaviside H functions [Kausel 2006]:

$$\boldsymbol{u}(\boldsymbol{r}, t) = \Big[A(r,t)\,\boldsymbol{I} + B(r,t)\,\boldsymbol{r}\boldsymbol{r}^\top\Big]\boldsymbol{f} \equiv \boldsymbol{D}(\boldsymbol{r}, t)\,\boldsymbol{f}, \tag{4a}$$

$$A(r,t) = \frac{1}{4\pi r}\left\{\frac{\delta(r-\beta t)}{\beta^2} + \frac{t}{r^2}\Big[H(r-\alpha t) - H(r-\beta t)\Big]\right\}, \tag{4b}$$

$$B(r,t) = \frac{1}{4\pi r^3}\left\{\frac{\delta(r-\alpha t)}{\alpha^2} - \frac{\delta(r-\beta t)}{\beta^2} - \frac{3t}{r^2}\Big[H(r-\alpha t) - H(r-\beta t)\Big]\right\}. \tag{4c}$$

where $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{c}$ is the relative position vector from the load center $\boldsymbol{c}$ to an observation point $\boldsymbol{x}$ at rest, and $r = \|\boldsymbol{r}\|$ is its norm. We refer to $\boldsymbol{D}(\boldsymbol{r}, t)$ as the Green's function for linear elastodynamics, which determines a 3×3 matrix mapping the force vector $\boldsymbol{f}$ at $\boldsymbol{c}$ to the displacement vector $\boldsymbol{u}$ at the relative position $\boldsymbol{r}$ and time t. Note that the first term in $\boldsymbol{D}(\boldsymbol{r}, t)$ is a radial scaling factor, while the second term modifies the elastic response based on the alignment of the relative position vector $\boldsymbol{r}$ to the force vector $\boldsymbol{f}$. We also point out that $\boldsymbol{u}(\boldsymbol{r}, t)$ is zero for any $t < 0$, since it precedes the activation of the impulse load $\boldsymbol{b}$. Importantly, notice the presence of singularities demarcating the elastic wave crests and the load center $\boldsymbol{c}$. Figure 2 (left) illustrates a time slice of (4a) at a fixed point $\boldsymbol{r}$.

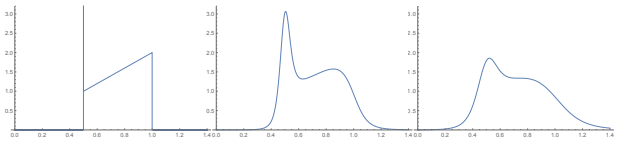


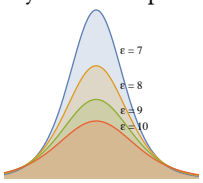
Fig. 2. **Regularization:** The plots show the displacement norm $\|\boldsymbol{u}(\boldsymbol{r}, t)\|$ as a function of time t at a fixed position $\boldsymbol{r} = \boldsymbol{e}_1$, with $\nu = 1/3$, $\mu = 1$, and $\boldsymbol{f} = \boldsymbol{e}_1$. The impulse dynamic Kelvinlet converges to the singular solution as $\varepsilon \to 0$ (see [Kausel 2006] p.51 for additional plots of the singular solution).

4 DYNAMIC KELVINLETS

The singularities introduced by the point load makes the fundamental solution of elastodynamics (4) unstable for numerical computations. To address this issue, we describe next a regularization scheme that generates finite and differentiable elastic deformations. First, we present a volumetric regularization of the body load and compute its Helmholtz decomposition. We then derive the analytical solutions of (2) in response to these load potentials. Finally, we construct dynamic Kelvinlets in closed-form by combining the elastic wave potentials and their derivatives.

4.1 Regularized Impulse Response

In order to form finite and localized elastic waves, we propose to replace the concentrated point load by a spatially smooth impulse $\boldsymbol{b}(\boldsymbol{r}, t) = \boldsymbol{f}\,\rho(r)\,\delta(t)$ with force vector $\boldsymbol{f}$ and normalized density function $\rho(r)$ distributed around the load center $\boldsymbol{c}$ by a radial scale $\varepsilon > 0$. Similar to [Cortez et al. 2005], we define a regularization function $(\cdot)_\varepsilon \equiv \left[(\cdot)^2 + \varepsilon^2\right]^{1/2}$. Using the regularized distance r_ε, we set the normalized density function to (see inset)



$$\rho(r) = \frac{15\varepsilon^4}{8\pi}\frac{1}{r_\varepsilon^7}. \tag{5}$$

Note that this regularized load $\boldsymbol{b}(\boldsymbol{r}, t)$ is a simple time extension of the elastostatic load $\boldsymbol{f}\,\rho(r)$ used in [de Goes and James 2017].

Due to the radial symmetry of the density function ρ, the Helmholtz decomposition of the impulse load $\boldsymbol{b}$ simplifies to:

$$\begin{cases} \Delta\psi = \nabla\cdot\boldsymbol{b} & \Rightarrow \quad \psi(\boldsymbol{r}, t) = -\mathcal{V}(r)\,\delta(t)\,\boldsymbol{f}^\top\boldsymbol{r}, & \text{(6a)} \\ \Delta\Psi = -\nabla\times\boldsymbol{b} & \Rightarrow \quad \Psi(\boldsymbol{r}, t) = -\mathcal{V}(r)\,\delta(t)\,\boldsymbol{f}\times\boldsymbol{r}. & \text{(6b)} \end{cases}$$

Here, the function $\mathcal{V}$ corresponds to a scalar potential field derived from ρ. As detailed in Appendix A, the displacement potentials ϕ and Φ generated by (6) via (2) present a similar form:

$$\begin{cases} \phi(\boldsymbol{r}, t) = \mathcal{U}_\alpha(r, t)\,\boldsymbol{f}^\top\boldsymbol{r}, & \text{(7a)} \\ \Phi(\boldsymbol{r}, t) = \mathcal{U}_\beta(r, t)\,\boldsymbol{f}\times\boldsymbol{r}, & \text{(7b)} \end{cases}$$

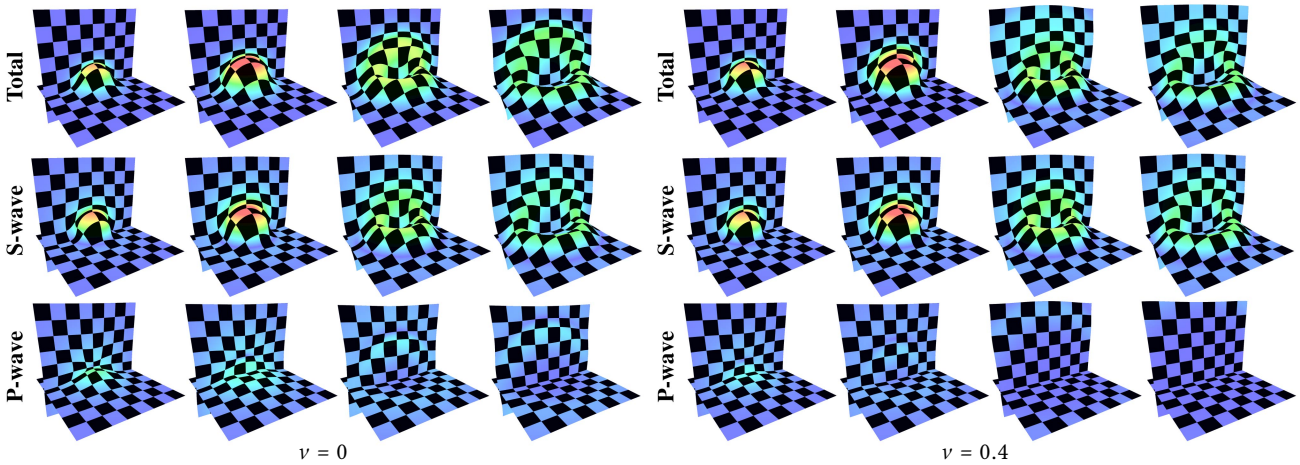


Fig. 3. **Elastic Waves:** Dynamic Kelvinlets offer deformations with elastic waves and interactive volume control. Top row shows animation sequences generated by a single impulse dynamic Kelvinlet with vertical force vector and different Poisson ratios. Pseudo-colors display a heat map of the time evolution of the displacement norm on two orthogonal planes. Observe that the plane aligned to the force vector depicts only texture deformations, while the perpendicular plane contains ripple-like dynamics. Middle row shows the contribution of the S-wave, which is incompressible and thus independent of the Poisson ratio. Bottom row indicates the contribution of the P-wave, which produces spherical blasts with speed and intensity proportional to the Poisson ratio.

where $\mathfrak{U}_\gamma$ (with $\gamma = \alpha$ or β) is a scalar pseudo-potential that quantifies the difference of an auxiliary function $\mathcal{W}$ evaluated at the radius r traveled forward and backwards by γt (see Appendix B):

$$\mathfrak{U}_\gamma(r,t) = \frac{1}{16\pi\gamma r^3}\Big[\mathcal{W}(r, r+\gamma t) - \mathcal{W}(r, r-\gamma t)\Big], \tag{8a}$$

$$\mathcal{W}(r,s) = \frac{1}{s_\varepsilon}\left(2s^2+\varepsilon^2-3rs\right) + \frac{1}{s_\varepsilon^3} r\, s^3. \tag{8b}$$

Equipped with the potentials ϕ and Φ in (7), the solution of elastodynamics (1) associated with a regularized impulse load (5) can be expressed in closed-form by:

$$\boxed{\boldsymbol{u}(\boldsymbol{r},t) = \Big[\mathcal{A}(r,t)\,\boldsymbol{I} + \mathcal{B}(r,t)\,\boldsymbol{r}\boldsymbol{r}^\top\Big]\boldsymbol{f} \equiv \mathcal{D}(\boldsymbol{r},t)\boldsymbol{f},} \tag{9a}$$

$$\mathcal{A}(r,t) = \mathfrak{U}_\alpha(r,t) + 2\mathfrak{U}_\beta(r,t) + r\,\partial_r\mathfrak{U}_\beta(r,t), \tag{9b}$$

$$\mathcal{B}(r,t) = \Big(\partial_r\mathfrak{U}_\alpha(r,t) - \partial_r\mathfrak{U}_\beta(r,t)\Big)/r. \tag{9c}$$

We point the reader to Appendix C for the closed-form expression of $\partial_r\mathfrak{U}_\gamma$. The solution in (9) is the building block of our method, and we name it the *impulse dynamic Kelvinlet*.

Figure 2 shows the influence of the regularizer ε to the impulse dynamic Kelvinlet by plotting a time-slice of $\boldsymbol{u}$ for a wedge of values of ε. Observe that our regularized solution removes the undesirable singularities at the wave crests, while reproducing the singular case as the radial scale ε approaches zero. Near the load center $\boldsymbol{c}$, (9) may seem problematic due to the $1/r$ terms, however, the asymptotic analysis of $\boldsymbol{u}$ as $r \to 0$ reveals that

$$\lim_{r\to 0}\boldsymbol{u}(\boldsymbol{r},t) = \frac{5t\varepsilon^4}{8\pi}\left(\frac{1}{(\alpha t)_\varepsilon^7} + \frac{2}{(\beta t)_\varepsilon^7}\right)\boldsymbol{f}, \tag{10}$$

where $(\alpha t)_\varepsilon$ and $(\beta t)_\varepsilon$ indicate the regularization of αt and βt respectively. Therefore, our regularized displacements $\boldsymbol{u}(\boldsymbol{r},t)$ are finite and differentiable for any time t and for any relative position $\boldsymbol{r}$. We further point out that our solution is feasible even in the incompressible regime ($\nu = 1/2$), with the P-wave contribution collapsing to zero, thus ensuring a divergence-free deformation.

As illustrated in Figure 3, a single impulse dynamic Kelvinlet is sufficient to create a volumetric ripple dynamics emanated from the load center $\boldsymbol{c}$, starting at time zero, and with a load support of size ε. Given a force vector $\boldsymbol{f}$, we can compute this temporal elastic deformation *analytically* by simply evaluating the displacements $\boldsymbol{u}(\boldsymbol{x}-\boldsymbol{c},t)$ for every point $\boldsymbol{x} \in \mathbb{R}^3$ and at any time t. By doing so, our formulation bypasses the need of any time-stepping, spatial discretization, or numerical solve.

4.2 Regularized Push Response

The impulse dynamic Kelvinlets generate elastic deformations that settle back to rest as time progresses, similar to ripples. However, some tasks may require dynamics with permanent deformations, such as in animation editing or dynamic sculpts. To address these cases, we introduce a steady version of the dynamic Kelvinlet that constructs elastic waves while converging to a finite displacement. The key idea is to replace the temporal Dirac $\delta(t)$ in the regularized impulse load $\boldsymbol{b}(\boldsymbol{r},t)$ by a temporal Heaviside profile $H(t)$, thus forming a regularized push load $\widehat{\boldsymbol{b}}(\boldsymbol{r},t) = \boldsymbol{f}\,\rho(r)H(t)$. Intuitively, this modification makes the regularized push load drag the region nearby the load center $\boldsymbol{c}$ steadily, while the regularized impulse load drags and releases the region around $\boldsymbol{c}$ instantaneously.

Since the Heaviside function is the integral of Dirac functions, we can exploit the linearity of the elastic wave equation with respect to integration and compute the solution of (1) associated with $\widehat{\boldsymbol{b}}$ via the time integral of the impulse dynamic Kelvinlet:

$$\boxed{\begin{aligned}\widehat{\boldsymbol{u}}(\boldsymbol{r},t) &= \int_0^t \boldsymbol{u}(\boldsymbol{r},\tau)\,d\tau \\ &= \left[\left(\int_0^t \mathcal{A}(r,\tau)\,d\tau\right)\boldsymbol{I} + \left(\int_0^t \mathcal{B}(r,\tau)\,d\tau\right)\boldsymbol{r}\boldsymbol{r}^\top\right]\boldsymbol{f} \\ &= \Big[\widehat{\mathcal{A}}(r,t)\,\boldsymbol{I} + \widehat{\mathcal{B}}(r,t)\,\boldsymbol{r}\boldsymbol{r}^\top\Big]\boldsymbol{f} \equiv \widehat{\mathcal{D}}(\boldsymbol{r},t)\boldsymbol{f}.\end{aligned}} \tag{11}$$

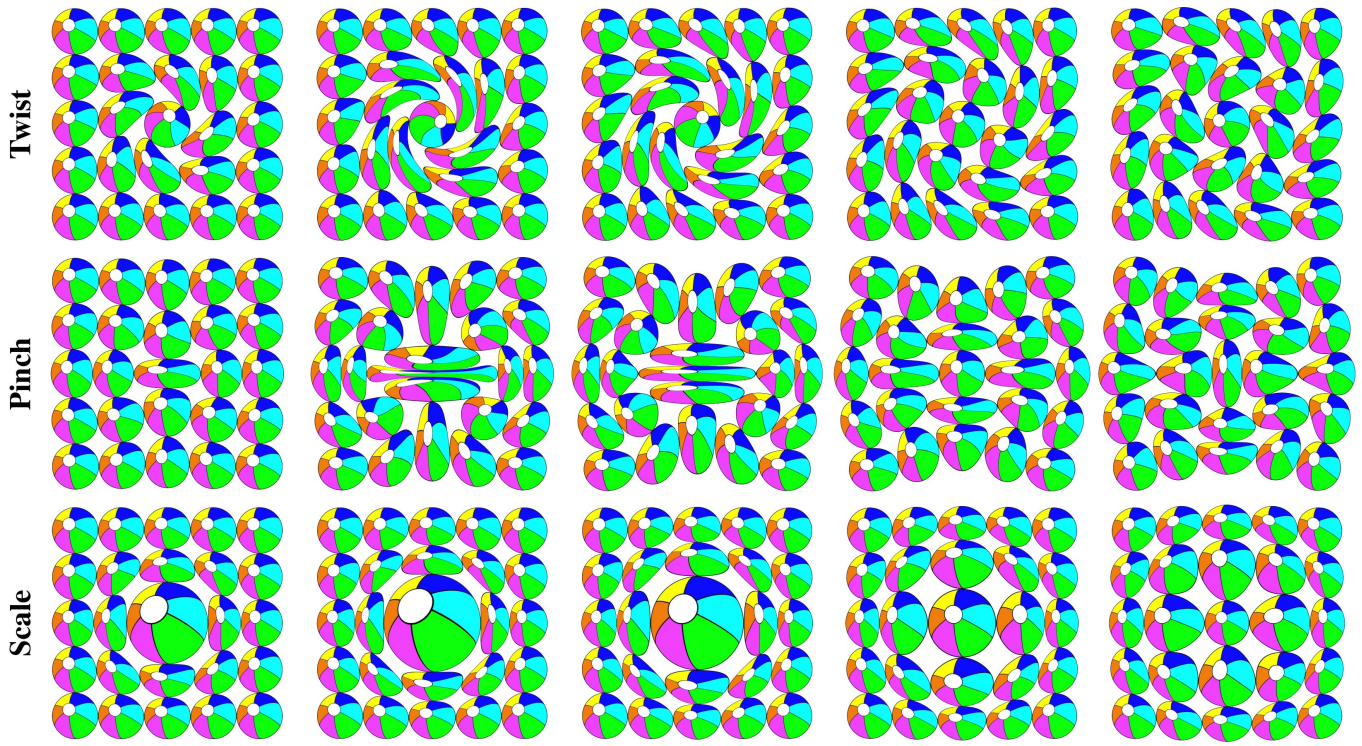


Fig. 4. **Affine-Impulse Dynamic Kelvinlets:** These examples show animation sequences of image deformations with elastic waves computed using tri-scale affine-impulse dynamic Kelvinlets, with force matrices assigned to a twist (top), a pinch (middle), and a uniform scale (bottom). We set the image size to 10×10, the Poisson ratio to $\nu = 0.45$, the stiffness to $\mu = 3.5$, and the radial scale to $\varepsilon = 4$.

We refer to the resulting displacement field $\widehat{\boldsymbol{u}}$ as the *push dynamic Kelvinlet*. In order to evaluate the scalar functions $\widehat{\mathcal{A}}$ and $\widehat{\mathcal{B}}$ in (11), we reuse (9b) and (9c), respectively, but with the pseudo-potential $\mathcal{U}_\gamma$ now substituted by its time integral, which yields

$$\widehat{\mathcal{U}}_\gamma(r,t) = \frac{1}{16\pi\gamma^2 r^3} Q\left(r, r+\gamma t, r-\gamma t\right), \tag{12a}$$

$$Q(r,s,w) = 2\frac{r^3}{r_\varepsilon} - sw\left(\frac{s}{s_\varepsilon}+\frac{w}{w_\varepsilon}\right) + \varepsilon^2(s-r)\left(\frac{1}{s_\varepsilon}-\frac{1}{w_\varepsilon}\right), \tag{12b}$$

where Q is a new auxiliary function that rearranges and simplifies the time integral of $\mathcal{U}_\gamma$. Similar to the impulse dynamic Kelvinlet, we can employ asymptotic analysis to confirm that the push displacements are finite and differentiable as $r \to 0$, yielding

$$\lim_{r\to 0} \widehat{\boldsymbol{u}}(\boldsymbol{r},t) = \frac{\varepsilon^4}{8\pi}\left[\frac{1}{\alpha^2}\left(\frac{1}{\varepsilon^5}-\frac{1}{(\alpha t)_\varepsilon^5}\right)+\frac{2}{\beta^2}\left(\frac{1}{\varepsilon^5}-\frac{1}{(\beta t)_\varepsilon^5}\right)\right]\boldsymbol{f}. \tag{13}$$

We also show in the supplemental material that the quasi-static state of the push dynamic Kelvinlet (i.e., the limit of $\widehat{\boldsymbol{u}}(\boldsymbol{r},t)$ as $t\to\infty$) reduces to the 3D regularized Kelvinlet introduced in [de Goes and James 2017]. Therefore, our result enriches volume sculpting tools with time-varying physically based deformations (see Figure 9).

4.3 Regularized Affine Response

So far we showed how to compute regularized elastic waves guided by a force vector that translates the load center. However, important physical phenomena, especially seismic blasts, are modelled via torsional and scaling loads (see, e.g., [Kausel 2006]). In order to capture these scenarios, we propose to extend the formulation of dynamic Kelvinlets to affine loads. Our approach follows the derivation of the locally affine regularized Kelvinlets [de Goes and James 2017], but now augmented with dynamics. Therefore, our deformations can be interpreted as a regularized version of dynamic elastic dipoles.

We first consider the affine extension of the impulse dynamic Kelvinlets. To this end, let the vectors $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3\}$ form an orthonormal bases spanning $\mathbb{R}^3$ and $\boldsymbol{F} = \left[F_{ij}\right]$ be a 3×3 matrix. We then define a regularized affine-impulse load $\widetilde{\boldsymbol{b}}$ as a linear combination of the spatial derivative of regularized impulse loads, i.e.,

$$\widetilde{\boldsymbol{b}}(\boldsymbol{r},t) = \delta(t)\,\boldsymbol{F}\,\nabla\rho(r) = \sum_{ij} F_{ij}\,\boldsymbol{e}_j^\top\,\nabla[\,\delta(t)\,\rho(r)\,\boldsymbol{e}_i\,]. \tag{14}$$

Since the elastic wave equation is linear with respect to differentiation, the solution of (1) associated with $\widetilde{\boldsymbol{b}}$ is

$$\widetilde{\boldsymbol{u}}(\boldsymbol{r},t) = \sum_{ij} F_{ij}\,\boldsymbol{e}_j^\top\,[\,\nabla\mathcal{D}(\boldsymbol{r},t)\,\boldsymbol{e}_i\,]. \tag{15}$$

By computing the derivatives of $\mathcal{D}$ (see Appendix C), we obtain the displacement field $\widetilde{\boldsymbol{u}}$ in terms of the force matrix $\boldsymbol{F}$:

$$\begin{aligned}\widetilde{\boldsymbol{u}}(\boldsymbol{r},t) = &\left[\frac{1}{r}\partial_r\mathcal{A}(r,t) - \mathcal{B}(r,t)\right]\boldsymbol{F}\boldsymbol{r}\\ &+\mathcal{B}(r,t)\left[\boldsymbol{F}+\boldsymbol{F}^\top+\operatorname{tr}(\boldsymbol{F})\,\boldsymbol{I}\right]\boldsymbol{r}+\frac{1}{r}\partial_r\mathcal{B}(r,t)\,(\boldsymbol{r}^\top\boldsymbol{F}\boldsymbol{r})\,\boldsymbol{r}.\end{aligned} \tag{16}$$

We name this matrix-based solution of elastodynamics the *affine-impulse dynamic Kelvinlet*. Note that the first term in (16) is simply the affine transformation $\boldsymbol{F}\boldsymbol{r}$ scaled by a time-varying radial factor, while the other two terms involve symmetric affine transformations.

We also point out that the resulting displacements are finite and differentiable, collapsing to zero as $r \to 0$. We can further construct specialized versions of the affine-impulse dynamic Kelvinlet by setting $\boldsymbol{F}$ to different types of matrices, as displayed in Figure 4.

Twisting: In the case of a skew-symmetric matrix, we can associate $\boldsymbol{F}$ to a vector $\boldsymbol{q}$ via the cross product matrix, i.e., $\boldsymbol{F} \equiv [\boldsymbol{q}]_\times$ where $[\boldsymbol{q}]_\times \boldsymbol{r} = \boldsymbol{q} \times \boldsymbol{r}$, then (16) simplifies to a twist deformation

$$\boxed{\boldsymbol{t}(\boldsymbol{r}, t) = \left[\frac{1}{r}\partial_r \mathcal{A}(r,t) - \mathcal{B}(r,t)\right] \boldsymbol{q} \times \boldsymbol{r}.} \tag{17}$$

By analyzing the gradient of (17), one can verify that its symmetric part is trivial for any $(\boldsymbol{r}, t)$. Consequently, this displacement field has zero divergence and defines a volume preserving deformation.

Scaling: Another type of affine-impulse dynamic Kelvinlet can be generated by a force matrix of the form $\boldsymbol{F} = s\,\boldsymbol{I}$, where s is a scalar. In this case, (16) reduces to a scaling deformation

$$\boxed{\boldsymbol{s}(\boldsymbol{r}, t) = \left[4\mathcal{B}(r,t) + \frac{1}{r}\partial_r \mathcal{A}(r,t) + r\partial_r \mathcal{B}(r,t)\right] s\,\boldsymbol{r},} \tag{18}$$

where positive values of s represent dilations, and negative values depict contractions. After some algebraic manipulation, we can also verify that $\boldsymbol{s}(\boldsymbol{r}, t) = 0$ for incompressible elastic materials.

Pinching: The last type of affine-impulse dynamic Kelvinlet is constructed using a symmetric matrix $\boldsymbol{F}$ with zero trace, yielding

$$\boxed{\boldsymbol{p}(\boldsymbol{r}, t) = \left[\frac{1}{r}\partial_r \mathcal{A}(r,t) + \mathcal{B}(r,t)\right] \boldsymbol{F}\boldsymbol{r} + \frac{1}{r}\partial_r \mathcal{B}(r,t)\,(\boldsymbol{r}^\top \boldsymbol{F}\boldsymbol{r})\,\boldsymbol{r}.} \tag{19}$$

Similar to [de Goes and James 2017], the deformation generated by $\boldsymbol{p}(\boldsymbol{r}, t)$ compensates infinitesimal stretching in one direction by contractions in the other directions, thus resembling a pinch.

At last, we can repeat the approach used in Sec. 4.2 and compute the time integral of the affine-impulse dynamic Kelvinlets. The resulting displacements offer elastic waves combined with permanent affine deformations nearby the load center $\boldsymbol{c}$. We thus refer to them as *affine-push dynamic Kelvinlets*. The closed-form expressions for these solutions mimic (16)-(19), but with the terms $\mathcal{A}$ and $\mathcal{B}$ replaced by $\widehat{\mathcal{A}}$ and $\widehat{\mathcal{B}}$ respectively. Since the limit and the gradient

Fig. 5. **Denting:** We used a time series of bi-scale push dynamic Kelvinlets to animate the formation of denting with elastic waves on a car model made of multiple disconnected parts. Forces, radial scales, and wave speeds were set procedurally by painting masks over the car shell. ©Disney/Pixar

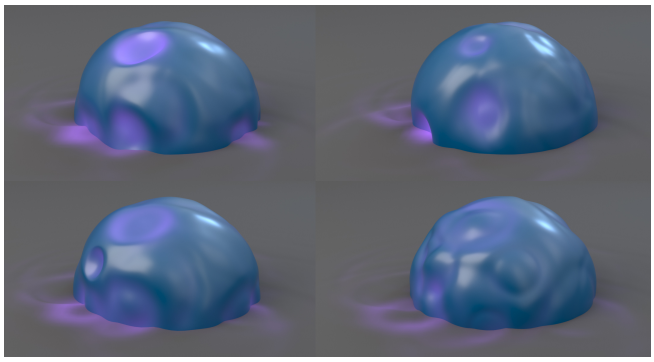


Fig. 6. **Time Series:** Multiple dynamic Kelvinlets can be combined to form a time series of elastic waves. This example shows a deformation sequence generated by displace- and affine-impulse dynamic Kelvinlets with various radial scale, volume compression, and wave speed set procedurally. The magenta shades indicate the elastic wavefronts propagated in time over the dome model and the planar ground.

operator commute, we can also show that the locally affine regularized Kelvinlets introduced in [de Goes and James 2017] correspond to the quasi-static state of the affine-push dynamic Kelvinlets. The accompanying video shows a side-by-side comparison of affine-push versus affine-impulse dynamic Kelvinlets.

5 COMPOUND ELASTIC WAVES

In this section, we describe the generation of elastic waves by multiple dynamic Kelvinlets. These compound deformations are of particular interest for imposing keyframe or pointwise constraints onto elastodynamics. For conciseness, we use $\boldsymbol{u}$ to denote any dynamic Kelvinlet, be it an impulse or push, with translational or affine load.

Superposition: We first consider the case with n regularized loads placed at $\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n\}$ and applied simultaneously. Due to the linearity of (1), the resulting displacement field is simply the superposition of dynamic Kelvinlets, i.e., $\sum_i \boldsymbol{u}_i(\boldsymbol{x} - \boldsymbol{c}_i, t)$. Therefore, the compound deformations can be evaluated for any point $\boldsymbol{x}$ and at any time t in parallel. When the loads are colocated at $\boldsymbol{c}_1$ but with different scales $\{\varepsilon_1, \ldots, \varepsilon_n\}$, the superposed displacements define multi-scale solutions, which can be used to design deformations with high-order spatial decay similar to the extrapolation scheme presented in [de Goes and James 2017]. Since dynamic Kelvinlets are linear in terms of force vectors $\boldsymbol{f}$ and matrices $\boldsymbol{F}$, we can further employ compound deformations to resolve displacement and gradient constraints. To this end, we include a vector/matrix-based dynamic Kelvinlet for each displacement/gradient constraint, and then precompute the load necessary to reproduce these pointwise constraints via a single linear solve.

Time Series: In order to construct a time series of elastic waves, we assign an activation timestamp t_i to each regularized load $\boldsymbol{b}_i$. The contribution of $\boldsymbol{b}_i$ to the net load applied to a point $\boldsymbol{x}$ is then defined with respect to a material point $\boldsymbol{x}_i$ representing the location of $\boldsymbol{x}$ at timestamp t_i, i.e., $\boldsymbol{b}_i(\boldsymbol{x}_i - \boldsymbol{c}_i, t - t_i)$. Using the linearity of (1) once more, the outcoming compound deformation is formed by the superposition of time-shifted dynamic Kelvinlets

$$\boldsymbol{x}(t) = \boldsymbol{x}_0 + \boldsymbol{u}(\boldsymbol{x}_0, t) = \boldsymbol{x}_0 + \sum_i \boldsymbol{u}_i(\boldsymbol{x}_i - \boldsymbol{c}_i, t - t_i)\,, \tag{20}$$

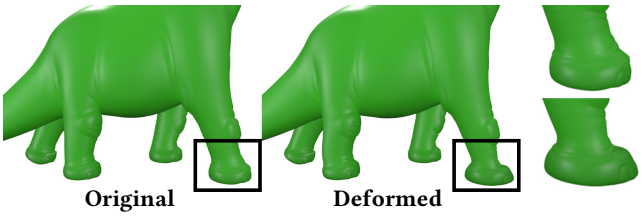


Fig. 7. **Impact Response:** Dynamic Kelvinlets are suitable to augment animations with secondary motions induced by impacts. This example shows the original frame of a foot-ground impact in a dinosaur walking cycle (left), versus the volume squashing generated by a pinch-impulse dynamic Kelvinlet applied at the point of impact (center). We also show close-ups of the foot before and after applying our method (right). ©Disney/Pixar

where $\boldsymbol{x}_0$ denotes the initial undeformed position of $\boldsymbol{x}$. Observe that the evaluation of (20) depends on the material points $\{\boldsymbol{x}_i\}$ corresponding to the position of $\boldsymbol{x}_0$ for every timestamp $\{t_i\}$. Given increasing timestamps $\{t_1 < t_2 < \ldots < t_n\}$, we also notice that each material point $\boldsymbol{x}_i$ is the accumulated displacement of the dynamic Kelvinlets precedent to t_i. For efficiency, we propose to precompute these material points sequentially via $\boldsymbol{x}_i = \boldsymbol{x}_0 + \boldsymbol{u}(\boldsymbol{x}_0, t_i)$. As a result, we can evaluate the displacement in (20) for multiple time values t in parallel. Figures 5 and 6 show compound deformations generated by a time series of superposed push and impulse dynamic Kelvinlets.

Transient Motions: In keyframed animation, it is often desirable to specify poses $\boldsymbol{y}_i$ at a few frames t_i and later add dynamics inbetween frames. We can achieve similar results by approaching keyframes as quasi-static constraints. More concretely, we consider keyframes $\boldsymbol{y}_i$ sculpted by regularized Kelvinlets [de Goes and James 2017], and use a compound of push dynamic Kelvinlets with keyframes $\boldsymbol{y}_i$ set to the material points $\boldsymbol{x}_i$ at their respective timestamps t_i. We can then compute the animated point $\boldsymbol{x}(t)$ at time $t \in [t_i, t_{i+1}]$ by including the transient motion determined by the difference between the dynamic Kelvinlets and their quasi-static solution, i.e.,

$$\boldsymbol{x}(t) = \boldsymbol{y}_i + \sum_{j \leq i} \left[\boldsymbol{u}_j(\boldsymbol{y}_j - \boldsymbol{c}_j, t - t_j) - \lim_{\tau \to \infty} \boldsymbol{u}_j(\boldsymbol{y}_j - \boldsymbol{c}_j, \tau) \right]. \quad (21)$$

Note that the transient motion in (21) reduces to zero as time progresses, thus reproducing the keyframes. Figure 9 displays a few frames generated by our keyframing approach.

6 RESULTS

We now detail our implementation of dynamic Kelvinlets, and discuss their performance and visual quality in a variety of scenarios. Please see the accompanying video for all wave animations, since these are particularly hard to depict in print.

Implementation: We implemented the dynamic Kelvinlets as a C++ plugin to Houdini [Side Effects 2018]. Our tool expects two inputs: a list of points $\boldsymbol{x}_0$ to be deformed and the specification of (possibly multiple) dynamic Kelvinlets. The latter can be set either procedurally or interactively, and includes the load center $\boldsymbol{c}_i$, radial scale ε_i, activation time t_i, force vector $\boldsymbol{f}_i$ and/or matrix $\boldsymbol{F}_i$, and the physical variables μ_i and ν_i that control wave speed and compressibility. We also support point constraints as an alternative to set the force parameters, and multi-scale extrapolation in order to refine the spatial falloff of the displacements. At any given time t, the resulting deformation is obtained by advecting each input point $\boldsymbol{x}$ through the displacement field $\boldsymbol{u}(\boldsymbol{x}, t)$ determined by the dynamic Kelvinlet. We compute this point advection using a fourth-order Runge-Kutta method, similar to [de Goes and James 2017; von Funck et al. 2006]. We avoid numerical issues by reverting to (10) and (13) when $r < 10^{-4}$, otherwise we use (9) and (11). Our implementation exploits multi-threading via Intel TBB [Reinders 2007], and takes an average of 3 ms per frame for scenes with a single dynamic Kelvinlet and 100k points, timed on a 2.3 GHz Intel Xeon E5-2699 with 18 cores. We point out that our elastic deformations can be performed at different frames simultaneously, with no time-stepping restriction, thus offering parallelization both in space and in time. To validate this feature, we deployed the dynamic Kelvinlets as a displacement shader using the VEX language supported by the Mantra renderer in Houdini. This is particularly desirable to deform subdivision surfaces, since the displacement evaluation can be executed per micropolygon in rendering time.

Examples: Figure 3 displays orthogonal slices of elastic waves associated with a single impulse dynamic Kelvinlet for distinct Poisson ratios. Observe that the vertical slice aligned to the force vector has deformations within the plane, which is visualized by the moving texture, while the deformations in the horizontal slice perpendicular to the force vector form ripples. This indicates the volumetric nature of elastic waves, in contrast to height-field water waves. Figure 4 illustrates animation sequences produced by different types of affine-impulse dynamic Kelvinlets. The results corresponding to affine-push dynamic Kelvinlets can be found in the supplemental video. Dynamic Kelvinlets are especially suitable to create volume blasts simulating impacts in animations. For instance, Figure 7 shows the jiggling caused by the feet impacts of a dinosaur in a walking cycle. For each impact point, we inserted a pinch-impulse dynamic Kelvinlet using (19) with compression along the direction of impact and uniform stretching on its perpendicular plane. Similarly, we combined a pinch-impulse and a displace-impulse dynamic Kelvinlet in Figure 1 to enrich a punch animation with volume oscillations resembling a shock wave. We also include in the accompanying video a simple rigid-body simulation augmented with impact waves produced by near-incompressible pinch-impulse dynamic Kelvinlets. The example in Figure 8 shows how to superpose dynamic Kelvinlets in order to conform loads to colliding geometries. In this case, we sampled each input curve with push dynamic Kelvinlets activated

Fig. 8. **Superposition:** By superposing dynamic Kelvinlets, we can compute deformations that conform to body loads of arbitrary spatial distributions. In this example, we constructed push dynamic Kelvinlets by sampling two input curves with radial scale set to the curve thickness and force vector aligned to the curve normals. Notice the localized squeezing caused by activating the curves at different timestamps. Model courtesy of Side Effects.

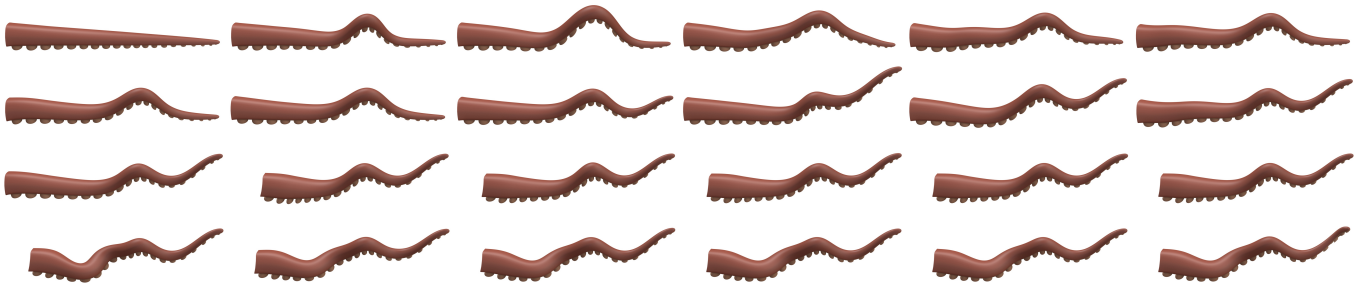


Fig. 9. **Keyframing:** Dynamic Kelvinlets can be used to create transient motions that interpolate keyframes. In this example, the right column shows keyframe poses sculpted via a series of regularized Kelvinlets [de Goes and James 2017]. The first two sculpts displace the center and the tip of the tentacle model, while the third and fourth poses pushes and rotates the base of the model respectively. The animation sequence with elastodynamic effects is then computed using a compound of incompressible push dynamic Kelvinlets with the keyframes set to quasi-static constraints, as described in § 5. ©Disney/Pixar

simultaneously and with radial scale set to the curve thickness. We present in Figure 6 a time series of vector- and matrix-based impulse dynamic Kelvinlets, and indicate the wavefronts with pseudo-colors. In Figure 5, we used a time series of bi-scale push dynamic Kelvinlets in order to create a denting effect in a car model with multiple disconnected pieces. In Figure 9, we keyframed poses of a tentacle model using regularized Kelvinlets [de Goes and James 2017], and then added transient dynamics inbetween frames via (21). We point the reader to the supplemental video for the animations of all these examples, in addition to interactive editing sessions.

Limitations: The simplicity of the dynamic Kelvinlets comes at the cost of several limitations. Our approach inherits the same trade-offs presented by regularized Kelvinlets [de Goes and James 2017]. In particular, our analytical solutions deform surfaces as if they were embedded in an infinite continuum. Therefore, the resulting dynamics is agnostic to any surface traction or boundary condition. This implies that the regularized elastic waves can only propagate spherically, with no knowledge of surface distances or boundary reflections. As an example, the inset shows elastic waves deforming a concave shape radially in response to a vertical force. The use of Dirac and Heaviside functions is also limiting, especially for loads with high-order temporal profiles. While multi-scale extrapolation provides control of the spatial falloff, we have not considered faster temporal falloffs computed, e.g., via damping. We finally note that our method is primarily useful for relatively modest dynamic effects to be added to preexisting animations, and it is not intended to replace advanced methods for physically based simulation.

Large Deformations: The linearized displacement model produced by dynamic Kelvinlets may cause distortion artifacts, especially for large rotational deformations. In previous works, linear-model artifacts have been addressed using mesh-based techniques such as warped stiffness [Müller and Gross 2004], modal warping [Choi and Ko 2005], or rotation-strain coordinates [Huang et al. 2011]. We instead alleviate linearization artifacts by adopting a mesh-independent line tracing approach that advects points along the analytical displacement field via a fourth-order Runge-Kutta method, similar to [de Goes and James 2017]. For large rotations, the results can be further improved using high-order line integral schemes [Nielson et al. 1997]. Figure 10 compares the point advection of an extreme twist-impulse dynamic Kelvinlet produced by a forward Euler advection versus line tracing with 20 substeps. The latter reproduces volume-preserving deformations with no visual artifacts.

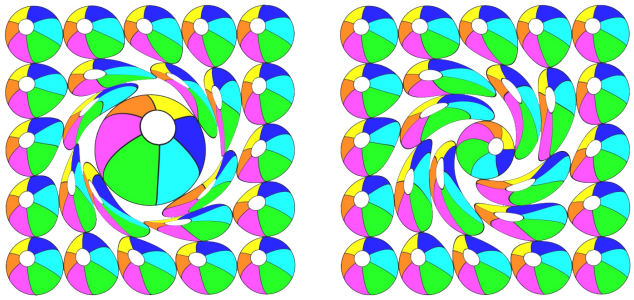


Fig. 10. **High-order Point Advection:** We avoid linearization artifacts by advecting points $\boldsymbol{x}$ through the dynamic Kelvinlet displacement field $\boldsymbol{u}(\boldsymbol{x}, t)$ evaluated at a given time t using multiple line tracing substeps. A twist deformation with no substeps shows visible rotational distortion (left), whereas 20 substeps virtually eliminates the artifact (right).

7 CONCLUSION

We have introduced dynamic Kelvinlets, a simple and interactive tool that allows animators to design elastic wave effects. Our approach is based on novel analytical solutions to the linear elastodynamic equations in the case of spatially regularized loads (such as impulses and steady forces) in infinite elastic domains. Since the solutions are expressed in closed form, we are able to compute mesh-independent deformations, without time-stepping restrictions, in a highly parallel manner. We have presented a number of practical animation applications in a production setting.

As future work, we are interested in the elastic response to more complex time profiles such as harmonic and ramp functions. We are also investigating the extension of our formulation to viscoelasticity by incorporating damping terms. While damping for impulse dynamic Kelvinlets is easily achieved with a temporal exponential

decay, the derivation of damped push solutions seems more intricate. Regularized fundamental solutions for strata and plates as well as restrictions to 2D domains are also of interest. Finally, we point out that dynamic Kelvinlets are well-suited to high-rate procedural applications in virtual and augmented reality.

ACKNOWLEDGMENTS

We are grateful to Vincent Serritella, Leon J. W. Park, and Stephen Marshall for their assistance with the examples in this work. We also thank Michael Rice, Matt Wong, Tolga Goktekin, and Michael Comet for discussions, and Mark Meyer and Tony DeRose for feedback on the manuscript.

REFERENCES

K. Aki and P.G. Richards. 1980. Quantitative Seismology: Theory and Methods. *W.H. Freeman, San Francisco* I (1980).
S. S. An, T. Kim, and D. L. James. 2008. Optimizing cubature for efficient integration of subspace deformations. In *ACM Trans. Graph.*, Vol. 27. ACM, 165.
A. Angelidis and K. Singh. 2007. Kinodynamic Skinning Using Volume-preserving Deformations. In *Proc. of the 2007 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*. 129–140.
Autodesk. 2016. Maya User Guide. (2016). https://autodesk.com/maya.
R. Barzel. 1997. Faking dynamics of ropes and springs. *IEEE Computer Graphics and Applications* 17, 3 (1997), 31–39.
S. Bouaziz, S. Martin, T. Liu, L. Kavan, and M. Pauly. 2014. Projective Dynamics: Fusing Constraint Projections for Fast Simulation. *ACM Trans. Graph.* 33, 4, Article 154 (2014).
J. A. Canabal, D. Miraut, N. Thürey, T. Kim, J. Portilla, and M. A. Otaduy. 2016. Dispersion Kernels for Water Wave Simulation. *ACM Trans. on Graph.* 35, 6 (2016).
M. G. Choi and H.-S. Ko. 2005. Modal Warping: Real-Time Simulation of Large Rotational Deformation and Manipulation. *IEEE Trans. on Visualization and Computer Graphics* 11, 1 (2005), 91–101.
R. Cortez. 2001. The Method of Regularized Stokeslets. *SIAM J. on Scientific Computing* 23, 4 (2001), 1204–1225.
R. Cortez, L. Fauci, and A. Medovikov. 2005. The method of regularized Stokeslets in three dimensions: Analysis, validation, and application to helical swimming. *Physics of Fluids* 17 (2005).
F. de Goes and D. L. James. 2017. Regularized Kelvinlets: Sculpting Brushes Based on Fundamental Solutions of Elasticity. *ACM Trans. Graph.* 36, 4, Article 40 (2017).
J. Dominguez. 1993. *Boundary Elements in dynamics.* WIT Press.
F. Hahn, S. Martin, B. Thomaszewski, R. Sumner, S. Coros, and M. Gross. 2012. Rig-space Physics. *ACM Trans. Graph.* 31, 4 (2012).
J. Huang, Y. Tong, K. Zhou, H. Bao, and M. Desbrun. 2011. Interactive Shape Interpolation Through Controllable Dynamic Deformation. *IEEE Trans. on Visualization and Computer Graphics* 17, 7 (2011), 983–992.
D. L. James and D. K. Pai. 2002. DyRT: dynamic response textures for real time deformation simulation with graphics hardware. In *ACM Trans. Graph.*, Vol. 21. ACM, 582–585.
S. Jeschke and C. Wojtan. 2015. Water wave animation via wavefront parameter interpolation. *ACM Trans. Graph.* 34, 3 (2015), 27.
S. Jeschke and C. Wojtan. 2017. Water wave packets. *ACM Trans. Graph.* 36, 4 (2017), 103.
M. Kass and J. Anderson. 2008. Animating oscillatory motion with overlap: wiggly splines. In *ACM Trans. Graph.*, Vol. 27. ACM, 28.
M. Kass and G. Miller. 1990. Rapid, Stable Fluid Dynamics for Computer Graphics. *SIGGRAPH* 24, 4 (1990), 49–57.
E. Kausel. 2006. *Fundamental Solutions in Elastodynamics: A Compendium.* Cambridge University Press.
T. Liu, A. W. Bargteil, J. F. O'Brien, and L. Kavan. 2013. Fast Simulation of Mass-spring Systems. *ACM Trans. Graph.* 32, 6, Article 214 (2013).
M. Müller and N. Chentanez. 2011. Solid Simulation with Oriented Particles. *ACM Trans. Graph.* 30, 4, Article 92 (2011).
M. Müller and M. H. Gross. 2004. Interactive Virtual Materials. In *Graphics Interface 2004*. 239–246.
M. Müller, B. Heidelberger, M. Teschner, and M. Gross. 2005. Meshless deformations based on shape matching. *ACM Trans. Graph.* 24, 3 (2005), 471–478.
A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson. 2006. Physically based deformable models in computer graphics. In *Computer graphics forum*, Vol. 25. 809–836.
G. Nielson, H. Hagen, and H. Müller. 1997. *Scientific Visualization.* IEEE Computer Society.
M. Pauly, D. K. Pai, and L. J. Guibas. 2004. Quasi-rigid Objects in Contact. In *Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*. Eurographics Association, 109–119.
A. Pentland and J. Williams. 1989. Good Vibrations: Modal Dynamics for Graphics and Animation. *SIGGRAPH Comput. Graph.* 23, 3 (1989), 207–214.
J. Reinders. 2007. *Intel Threading Building Blocks.* O'Reilly & Associates, Inc.
C. Schulz, C. von Tycowicz, H.-P. Seidel, and K. Hildebrandt. 2014. Animating Deformable Objects Using Sparse Spacetime Constraints. *ACM Trans. Graph.* 33, 4, Article 109 (2014).
C. Shen, T. Hahn, B. Parker, and S. Shen. 2015. Animation Recipes: Turning an Animator's Trick into an Automatic Animation System. In *ACM SIGGRAPH Talks*. ACM, Article 29.
Side Effects. 2018. Houdini Engine. (2018). http://www.sidefx.com.
G. G. Stokes. 1849. On the dynamical theory of diffraction. *Trans. Camb. Phil. Soc.* 9 (1849), 1–62.
J. Tessendorf. 2001. Simulating ocean water. *Simulating nature: realistic and interactive techniques. SIGGRAPH Courses* 1, 2 (2001), 5.
W. von Funck, H. Theisel, and H. P. Seidel. 2006. Vector field based shape deformations. *ACM Trans. on Graph.* 25, 3 (2006), 1118–1125.
W. von Funck, H. Theisel, and H. P. Seidel. 2007. Elastic Secondary Deformations by Vector Field Integration. In *Proc. of the Fifth Eurographics Symp. on Geometry Processing*. 99–108.
C. Yuksel, D. H. House, and J. Keyser. 2007. Wave particles. In *ACM Trans. Graph.*, Vol. 26. ACM, 99.

A POTENTIAL FIELDS

We now detail the derivation of the potential fields used in § 4.1. We start with an auxiliary potential field $\mathcal{R}$ induced by ρ:

$$-\Delta\mathcal{R}=\rho \Rightarrow \mathcal{R}(r)=\int_{\mathbb{R}}\frac{\rho(s)}{4\pi|r-s|}ds=\frac{1}{4\pi r_\varepsilon}\left[1+\frac{\varepsilon^2}{2r_\varepsilon^2}\right].$$

This integral is equivalent to "$\psi_\varepsilon(r)$" from [de Goes and James 2017]. We now define the potential $\mathcal{V}$ in (6) based on the gradient of $\mathcal{R}$:

$$\nabla\mathcal{R}(r)=\left[\frac{1}{r}\partial_r\mathcal{R}(r)\right]\boldsymbol{r}=-\mathcal{V}(r)\boldsymbol{r} \Rightarrow \mathcal{V}(r)=\frac{1}{4\pi r_\varepsilon^3}\left[1+\frac{3\varepsilon^2}{2r_\varepsilon^2}\right].$$

Using the Laplacian Green's function, we can verify (6a):

$$\psi(\boldsymbol{r},t)=\int_{\mathbb{R}^3}\frac{\nabla\cdot\boldsymbol{b}(\boldsymbol{s},t)}{4\pi\|\boldsymbol{r}-\boldsymbol{s}\|}d\boldsymbol{s}=\delta(t)\,\boldsymbol{f}^\top\nabla\mathcal{R}(r)=-\mathcal{V}(r)\,\delta(t)\,\boldsymbol{f}^\top\boldsymbol{r}.$$

We can then verify (7a) by rewriting (2a) as an integral with the Green's function of the wave equation:

$$\begin{aligned}\phi(\boldsymbol{r},t)&=\int_{\mathbb{R}^3}\int_{\mathbb{R}}\psi(\boldsymbol{s},t-\tau)\frac{\delta\left(\tau-\|\boldsymbol{r}-\boldsymbol{s}\|/\alpha\right)}{4\pi\alpha^2\|\boldsymbol{r}-\boldsymbol{s}\|}d\tau d\boldsymbol{s}\\&=\frac{-1}{4\pi\alpha^2}\int_{\mathbb{R}^3}\frac{\mathcal{V}(s)\,\delta\left(t-\|\boldsymbol{r}-\boldsymbol{s}\|/\alpha\right)\boldsymbol{f}^\top\boldsymbol{s}}{\|\boldsymbol{r}-\boldsymbol{s}\|}d\boldsymbol{s}=\mathfrak{V}_\alpha(r,t)\boldsymbol{f}^\top\boldsymbol{r}.\end{aligned}$$

A similar derivation can also be used to compute the expressions of Ψ in (6b) and Φ in (7b).

B PSEUDO-POTENTIAL

In this section, we describe how the closed-form expression of pseudo-potential $\mathfrak{V}_\alpha$ in (8a) was derived (c.f. the singular case in [Aki and Richards 1980]). We begin with the definition of the pseudo-potential $\mathfrak{V}_\alpha$ presented in the end of Appendix A:

$$\mathfrak{V}_\alpha(r,t)\,\boldsymbol{r}=\frac{-1}{4\pi\alpha^2}\int_{\mathbb{R}^3}\frac{\delta\left(t-\|\boldsymbol{r}-\boldsymbol{s}\|/\alpha\right)\mathcal{V}(s)\,\boldsymbol{s}}{\|\boldsymbol{r}-\boldsymbol{s}\|}d\boldsymbol{s}.$$

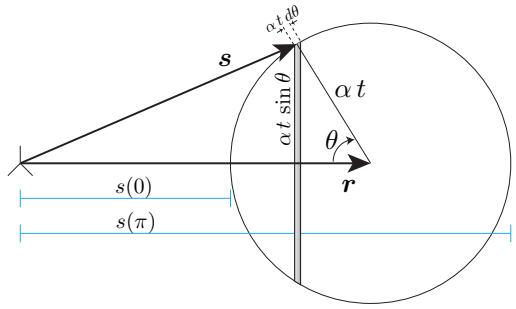


Fig. 11. **Spherical shell:** Notation for spherical shells in the case of $r > \alpha t$.

We can compute this integral by expanding it over spherical shells $S(\boldsymbol{r},\tau) = \{\boldsymbol{s} \in \mathbb{R}^3 : \alpha\tau = \|\boldsymbol{r}-\boldsymbol{s}\|\}$. Since $d\boldsymbol{s} = \alpha\, dS\, d\tau$, we have for $t > 0$:

$$\begin{aligned}\mathfrak{U}_\alpha(r,t)\,\boldsymbol{r} &= \frac{-1}{4\pi\alpha^2}\int_0^\infty \int_{S(\boldsymbol{r},\tau)} \left(\frac{\delta(t-\tau)}{\alpha\tau}\mathcal{V}(s)\,\boldsymbol{s}\right)\alpha\, dS\, d\tau \\ &= \frac{-1}{4\pi\alpha^2\, t}\left(\int_{S(\boldsymbol{r},t)} \mathcal{V}(s)\,\boldsymbol{s}\, dS\right).\end{aligned}$$

With the potential $\mathcal{R}$ from Appendix A, we can rewrite the spherical shell integral as the gradient of an auxiliary function $h(\boldsymbol{r},t)$:

$$-\int_S \mathcal{V}(s)\,\boldsymbol{s}\, dS = \int_S \nabla\mathcal{R}(s)\, dS = \nabla\int_S \mathcal{R}(s)\, dS = \nabla h(\boldsymbol{r},t),$$

where we have used the fact that $\boldsymbol{s} \equiv \boldsymbol{r} + \text{const}$ on S for fixed t. Using spherical coordinates θ and φ, we can integrate $\mathcal{R}$ through circular strips of width $\alpha\, t\, d\theta$ and circle radius $\alpha\, t \sin\theta$. Since the circle is at a constant distance $s(\theta)$ from the origin, the axial φ-integral is trivially 2π (see Figure 11). Therefore, the h-integral becomes

$$h(\boldsymbol{r},t) = \int_S \mathcal{R}(s)\, dS = 2\pi\alpha^2 t^2 \int_0^\pi \mathcal{R}(s)\, \sin\theta\, d\theta.$$

It follows from the law of cosines $s^2 = r^2 + \alpha^2 t^2 - 2r\alpha t\cos\theta$ that taking differentials on a shell of constant t gives $r\alpha t \sin\theta\, d\theta = s\, ds$. As a consequence, we obtain:

$$h(\boldsymbol{r},t) = \frac{2\pi\alpha t}{r}\int_{s(0)}^{s(\pi)} \mathcal{R}(s)\, s\, ds,$$

which we can evaluate using the indefinite integral identity:

$$\int \mathcal{R}(s)\, s\, ds = \frac{4s^2 + 3\varepsilon^2}{16\pi s_\varepsilon}.$$

One last wrinkle is that the minimum $s(0)$ and maximum $s(\pi)$ integral limits are nonsmoothly dependent on r and t, i.e., $s(\pi) = r + \alpha t$ and $s(0) = |r - \alpha t|$. However, the integral only involves quadratic terms and then the derivative discontinuity disappears, yielding:

$$h(r,t) = \frac{\alpha t}{8r}\left[\frac{4(r+\alpha t)^2 + 3\varepsilon^2}{(r+\alpha t)_\varepsilon} - \frac{4(r-\alpha t)^2 + 3\varepsilon^2}{(r-\alpha t)_\varepsilon}\right].$$

Putting it all together, we can write $\mathfrak{U}_\alpha(r,t)$ for $t > 0$ as:

$$\mathfrak{U}_\alpha(r,t)\,\boldsymbol{r} = \frac{1}{4\pi\alpha^2\, t}\nabla h(r,t) = \frac{1}{16\pi\alpha\, r^3}\left[\frac{4r^2}{\alpha t}\partial_r h(r,t)\right]\boldsymbol{r},$$

and, after algebraic simplification, we obtain (8a).

C DERIVATIVES

The affine-impulse dynamic Kelvinlets require the radial derivatives of the terms $\mathcal{A}$ and $\mathcal{B}$. Via direct differentiation, we have

$$\begin{aligned}\partial_r\mathcal{A}(r,t) &= \partial_r\mathfrak{U}_\alpha(r,t) + 3\partial_r\mathfrak{U}_\beta(r,t) + r\partial_{rr}\mathfrak{U}_\beta(r,t), \\ \partial_r\mathcal{B}(r,t) &= \frac{1}{r}\left[\partial_{rr}\mathfrak{U}_\alpha(r,t) - \partial_{rr}\mathfrak{U}_\beta(r,t) - \mathcal{B}(r,t)\right].\end{aligned}$$

These expressions depend on the derivatives of the pseudo-potential $\mathfrak{U}_\alpha$. For conciseness, we denote $r_+ = r + \alpha t$ and $r_- = r - \alpha t$. The first and second radial derivatives of $\mathfrak{U}_\alpha$ can then be written as:

$$\begin{aligned}\partial_r\mathfrak{U}_\alpha(r,t) &= \frac{1}{16\pi\alpha r^3}\left[\partial_r - \frac{3}{r}\right]\Big(\mathcal{W}(r,r_+) - \mathcal{W}(r,r_-)\Big), \\ \partial_{rr}\mathfrak{U}_\alpha(r,t) &= \frac{1}{16\pi\alpha r^3}\left[\partial_{rr} - \frac{6}{r}\partial_r + \frac{12}{r}\partial_{rr}\right]\Big(\mathcal{W}(r,r_+) - \mathcal{W}(r,r_-)\Big).\end{aligned}$$

Therefore, we need the derivatives of the auxiliary function $\mathcal{W}$. Note that $\partial_r\mathcal{W}(r,s) = \partial^1\mathcal{W}(r,s) + \partial^2\mathcal{W}(r,s)$, where $s = r \pm \alpha t$ and ∂^i is the partial derivative with respect to the i-th argument. After algebraic manipulation, we obtain:

$$\partial_r\mathcal{W}(r,s) = -3\varepsilon^4\left(\frac{r}{s_\varepsilon^5}\right), \quad \partial_{rr}\mathcal{W}(r,s) = -3\varepsilon^4\left(\frac{s_\varepsilon^2 - 5rs}{s_\varepsilon^7}\right).$$

For the affine-push dynamic Kelvinlets, we can take a similar approach and evaluate the radial derivatives of $\widehat{\mathcal{A}}$ and $\widehat{\mathcal{B}}$ by computing the derivatives of $\widehat{\mathfrak{U}}_\alpha$.

$$\begin{aligned}\partial_r\widehat{\mathfrak{U}}_\alpha(r,t) &= \frac{1}{16\pi\alpha^2 r^3}\left[\partial_r - \frac{3}{r}\right]\mathcal{Q}(r,r_+,r_-), \\ \partial_{rr}\widehat{\mathfrak{U}}_\alpha(r,t) &= \frac{1}{16\pi\alpha^2 r^3}\left[\partial_{rr} - \frac{6}{r}\partial_r + \frac{12}{r}\partial_{rr}\right]\mathcal{Q}(r,r_+,r_-).\end{aligned}$$

The derivatives of the auxiliary function $\mathcal{Q}$ can be computed using

$$\partial_r\mathcal{Q}(r,s,w) = \partial^1\mathcal{Q}(r,s,w) + \partial^2\mathcal{Q}(r,s,w) + \partial^3\mathcal{Q}(r,s,w),$$

with $s = r_+$ and $w = r_-$, which produces

$$\begin{aligned}\partial_r\mathcal{Q}(r,s,w) &= 2r^2\left(\frac{3}{r_\varepsilon} - \frac{r^2}{r_\varepsilon^3}\right) - 2r\left(\frac{s}{s_\varepsilon} + \frac{w}{w_\varepsilon}\right) - \varepsilon^2 r\left(\frac{s}{s_\varepsilon^3} + \frac{w}{w_\varepsilon^3}\right), \\ \partial_{rr}\mathcal{Q}(r,s,w) &= 2\left(\frac{2r}{r_\varepsilon} - \frac{s}{s_\varepsilon} - \frac{w}{w_\varepsilon}\right) + \varepsilon^2\left(\frac{2r}{r_\varepsilon^3} - \frac{s}{s_\varepsilon^3} - \frac{w}{w_\varepsilon^3}\right) + 3r\varepsilon^4\left(\frac{2}{r_\varepsilon^5} - \frac{1}{s_\varepsilon^5} - \frac{1}{w_\varepsilon^5}\right).\end{aligned}$$