# Stokes Coordinates

Yann Savoye
Robert Gordon University

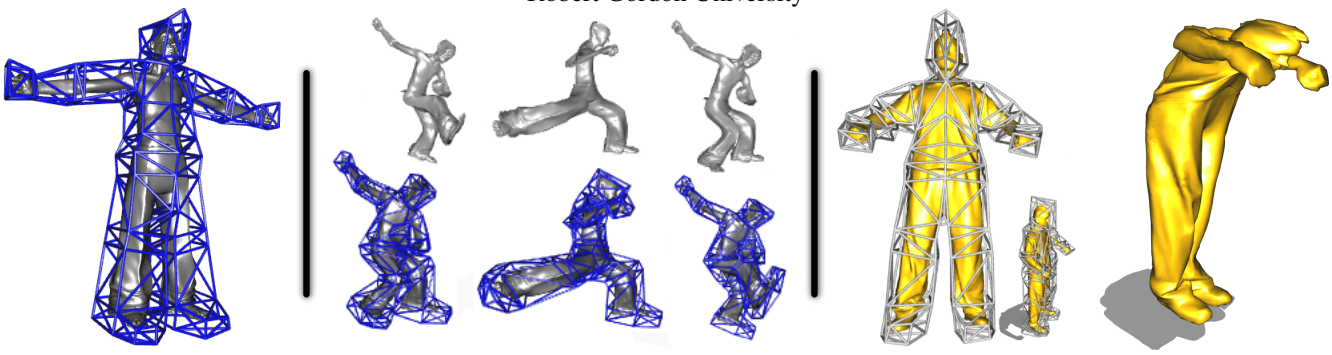

**Figure 1:** ***Stokes Cage Coordinates.*** *We compute our fluid-inspired Stokes Coordinates using an input undeformed model and its shape-aware cage (**left-hand side**). To obtain coordinates via vorticity transport, we derive the Stokes stream function using a compact second-order approximation with center-differencing. Given a collection of deformed cage poses, we output a sequence of deforming poses using our Stokes Coordinates (**middle**). To showcase the usefulness of our technique, we produce cartoon-style non-rigid deformation for a performance capture mesh. The resulting non-isometric deformation is quite fluid-like thanks to the local volume-preserving property since non-isometric deformation can be encoded using our proposed coordinates (**right-hand side**).*

## Abstract

Cage-based structures are reduced subspace deformers enabling non-isometric stretching deformations induced by clothing or muscle bulging. In this paper, we reformulate the cage-based rigging as an incompressible Stokes problem in the vorticity space. The key to our approach is a compact stencil allowing the expression of fluid-inspired high-order coordinates. Thus, our cage-based coordinates are obtained by vorticity transport as the numerical solution of the linearized Stokes equations. Then, we turn the incompressible creeping Newtonian flow into Stokes equations, and we devise a second-order compact approximation with center differencing for solving the vorticity-stream function. To the best of our knowledge, our work is the first to devise a vorticity-stream function formulation as a computational model for cage-based weighting functions. Finally, we demonstrate the effectiveness of our new techniques for a collection of cage-based shapes and applications.

## 1 Introduction

*" Empty your mind. You must be shapeless, formless, like water. When you put water in a cup, it becomes the cup. When you put water in a bottle, it becomes the bottle. When you put water in a teapot, it becomes the teapot. Water can flow and it can crash."*

— Bruce Lee, *The Warrior Within.*

**Animator-Oriented Deformers.** Subspace deformers are used by artists to generate life-like skin deformation. Furthermore, shape coordinates are essential for space-time shape modeling and geometry encoding as seen with reduced subspace cages. Technically speaking, cage deformers are closed polyhedral proxy meshes enclosing the given model to deform. In this work, our research problem is to define fluid-inspired cage-based coordinates allowing non-rigid boneless deformations with local preserving properties. This property seems difficult to reach exclusively in term of piecewise-rigid skeletal rigs.

**Fluid-inspired Coordinates.** Our motivation is to bring fluid dynamics formulation into cage-based deformation. Derived from Newton's second law of fluid motion, Navier-Stokes equations are fundamental in numerous physics problems to describe the dynamics of general flow for the fluid motion. Also, Stokes equations are of particular importance in solving physical problems by describing the dynamics of general fluid flows. Moreover, we rely on fluid dynamics to offer a theoretical understanding of how to compute cage coordinates. In this paper, we reformulate cage-based rigging as an incompressible Stokes problem. Thus, we establish a strong intuition for why cage basis functions should be framed in terms of fluid flow, most likely by neglecting non-linear terms. Our motivation is to rely on fluid vorticity transport principles instead of using the traditional heat diffusion to obtain high-order cage-based coordinates. Our fluid-inspired intuition is justified by the mass preserving property leading to the stable computation of well-localized weights for cages. In this work, we study how the Stokes formulation can be suitable for cage weighting functions since the governing Stokes equation transports values stored at the cage vertices across the enclosed mesh with a vorticity propagation behavior.

**Compact-Stenciled Stokes Coordinates.** At the heart of our space based rigging approach, we cast weight coordinates as the solution to the steady-state fluid flow, governed by the Stokes fluid model. We introduce a novel deforming weighting scheme by considering the numerical approximation of the Stokes equations with center differencing. The solutions to the Stokes equations are called *Stokes Coordinates* and they are used as weighting terms for the cage-based shape deformation. In contrast to previous techniques that rely on heat transfer in the solid, our approach differs by taking advantage of the conservation of mass principle offered by Stokes fluids.

**Our Contributions.** Our central contribution is the mathematical bridge between vorticity stream function formulation and cage-based coordinates. Our mathematical formulation brings generality to solve the problem. To make the computation amendable, we devise two key derivations: *Stokes Linearization* and *Biharmonization*. As our *Stokes Coordinates* are solutions to Stokes flows at steady state, we develop an iterative *Volumetric Stokes Weights Solver* to discretize vorticity transport. To speed up the computation and facilitate the boundary condition management, we directly formulate a compact approximation for our Stokes-wise biharmonic operator by employing a compact cell neighborhood stencil.


**Figure 2:** ***Overview.*** *We propose a five-stage pipeline for the computation of our Stokes Coordinates techniques dedicated to cage-based encoding, deformation and animation.*

## 2 Related Works

In this section, we first review relevant works related to our *Stokes Coordinates* for cages covering the following research areas: *physic-inspired rigging*, *cage-based subspace coordinates*, *manifold harmonicity* and *diffusion-based fluid dynamics*.

**Subspace Coordinates.** In contrast with Baran *et al.* [Baran and Popović 2007], we replace the skeletal-based representation with a fully deformable cage to achieve extreme non-rigid local deformation. Cage-based deformation is an important tool in *Geometry Processing*. Purely geometric deformers are represented by generated coarse cages [Le and Deng 2017] offering reusability and abstraction [Ju et al. 2008]. In addition, deformation fields are also classical strategy to deform a shape with reduced control [Jacobson et al. 2012a; Xu et al. 2009; Shen et al. 2010]. Typically, the coarse cage deformers are augmented with coordinates having closed-form [Ju et al. 2005] and quasi-conformal properties [Lipman et al. 2008]. Interestingly, *Harmonic Coordinates* [Joshi et al. 2007; Weber et al. 2012] overcome the lack of smoothness of *Mean Value Coordinates* [Ju et al. 2005] but suffer from non-closeness form. To remedy this problem, Lipman *et al.* [Lipman et al. 2008] propose one of the first closed-form and quasi-conformal coordinate systems for cages using Green's equation. Also, Ben-Chen *et al.* [Ben-Chen et al. 2009] introduced variational harmonic maps for space deformation as higher-order barycentric coordinates [Ju et al. 2008] and Jacobson *et al.* [Jacobson et al. 2012b] propose a harmonic shape deformation without local extrema. To reach compactness, Zhang *et al.* [Zhang et al. 2014] prefer to define a family of local barycentric coordinates using a small set of control points. Recently, researchers mixed various coordinates into a unique multi-coordinates system [García et al. 2013], and to develop biharmonic functions with gradient constraints [Li et al. 2013]. More recently, Budninskiy *et al.* [Budninskiy et al. 2016] introduces Voronoi-based coordinates as generalized barycentric coordinates on convex polytopes. Our work is also related to recent effort of Wang *et al.* [Wang et al. 2015] introducing non-negative weight for producing as-rigid-as possible deformation. Even if our work is focused on harmonic-type control [Joshi et al. 2007], our key insight is the introduction of the fluid-inspired generalization of the *Biharmonic Coordinates* [Jacobson et al. 2011; Weber et al. 2012] having local mass-conserving properties. Our technique also avoids complex black-box numerical solver while offering an easy-to-implement solution comparing to previous works.

**Physic-inspired Deformation.** Recent years have seen an important collection of physics-inspired rigging methods for elastic character animation [Angelidis and Singh 2007; von Funck et al. 2007]. Even if physics-inspired rigging is a well-explored field for elastic character animation [McAdams et al. 2011], our work introduces flow dynamics to compute cage-based coordinates. Our method exploits linearity [Capell et al. 2005] and domain decomposition [Kim and James 2011] to generate physics-based character skinning, but we prefer to rely on a skin-detached reduced subspace domain rather than a direct forces system.

Elasticity is a useful theory employed in the works of McAdams *et al.* [McAdams et al. 2011] and the *elasticity-inspired* deformers of Kavan *et al.* [Kavan and Sorkine 2012]. Earlier, Capell *et al.* [Capell et al. 2005] proposed a rigging system based on a force-driven elastic linearization scheme. Similarly, we are inspired by physical dynamics and linearity, but we prefer to rely on a detached cage deformer instead of a force-based system. More recently, Hahn *et al.* [Hahn et al. 2012] formalized a physically-based rig-space to add rich physical motions by defining equations of motions in the deformation subspace. Also, we take inspiration from Wang *et al.* [Wang et al. 2013] that built a connection between harmonic parametrization and electrostatics and Zheng *et al.* [Zheng and James 2009] to benefit from the harmonicity in fluid dynamics. In our work, we take advantage of a domain-decomposition method similar to Kim *et al.* [Kim and James 2011] to generate physics-based character skinning. However, we define our decomposition inside a reduced subspace domain, restricted to all cage handles.

**Fluid-inspired Rigging.** Classically, heat diffusion is used for the transfer process but this technique does not preserve the mass. However, diffused scalar fields can be obtained by mass-preserving vorticity transport using the fluid dynamics. Hence, we are interested in deriving cage-based coordinates from Stokes equation and derived streamfunction rather than relying on fluid simulation. Thus, we define handle-aware volumetric fields controllable by cage vertices. Our work differs from classical studies about fluid dynamics in Computer Graphics in the sense that we exploit flow dynamics to compute cage-based coordinates away from its original context. Unfortunately, fluid dynamics require a heavy use of finite element method to solve non-linear equation [Turner and Mazzone 1999; Shang et al. 2011; Rannacher 2000; Cai et al. 2010]. Consequently, we are interested in linearizing ideas of Jovanovic *et al.* [Jovanovic and Bamieh 2001], coupled with a highly reliable discretization [Fishelov et al. 2011] in the context of geometric subspaces [Barbič and James 2005] represented by deformable cages. Analogous to Xu *et al.* [Xu and He 2013] and Shang *et al.* [Shang 2013], we also employ an iterative method to obtain the steady state in our work, but we design our solver for a simplified Stokes formulation. Finally, Ando *et al.* [Ando et al. 2015] derive the stream function for incompressible fluids, but the formulation seems not directly applicable to cage-based applications. In contrast with Feng *et al.* [Feng and Warren 2012], we define a discrete Stokes-wise biharmonic operator using a finite-difference solution [Greenspan and Schultz 1972]. Finally, preconditioners for incompressible Stokes problems [Segal et al. 2010; ur Rehman et al. 2010] could also improve the accuracy of numerical results.

## 3 Stokes Cage Coordinates

In this section, we develop our novel mathematical derivation of streamline-vorticity and its discretization using finite difference method to obtain our compact-stenciled *Stokes Coordinates*. Then, we show several applications like cage-based shapes encoding, deformation and reusing (see Figure 1).

**Overview.** We propose an overview of our five-stage derivation strategy illustrated in Figure 2. We assume the common assumption for incompressible creeping Newtonian flow [Karoly and Laszlo 2013] with constant density and constant viscosity. First, we start from the incompressible Navier-Stokes equation as a formulation

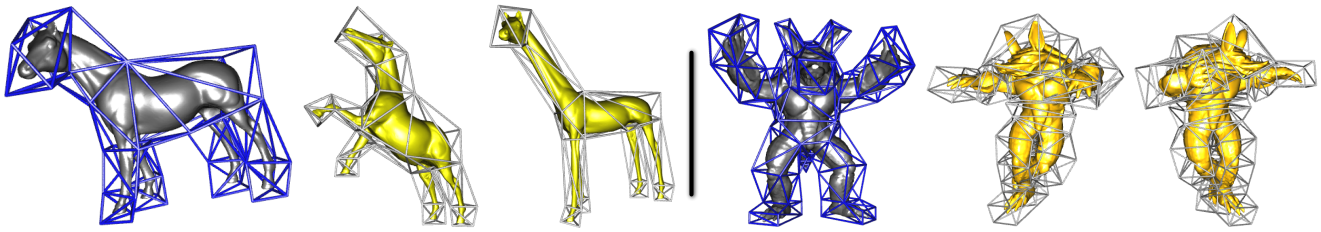

**Figure 3:** ***Cage-based Animation.*** *Taking the* `Horse` *model, various deformed cages and our Stokes Coordinates as input, we output a sequence of stretched cage-based shapes are obtained with our Stokes Coordinates and cages interpolation (**left-hand side**). The undeformed pose of the* `Armadillo` *model and two deformed poses obtained with our Stokes Coordinates (**right-hand side**).*

for the fluid transport. In the second step, we linearize the Stokes flow by neglecting the convective and by making viscous forces dominant. In the third step, we derive a Stokes-wise biharmonic form for Stokes equations, and we solve this equation for each cage handle. Various assumptions made in the derivation are valid for the purpose of cage-based deformation since cage weights computation only require to be transported. Then, neglecting these physical terms is permissible. In the fourth step, we formulate a compact stencil to derive the Stokes-wise biharmonic operator using a second-order approximation with center-differencing. In steady state, we obtain our *Stokes Coordinates* using an iterative volumetric solver to discretize the biharmonic equation of the stream function. Finally, we reuse *Stokes Coordinates* for cage-based geometric applications.

**Navier-Stokes Flows.** The velocity-pressure formulation for fluid is defined as follows by Childress [Childress 2007]:

$$Re\left(\frac{\partial \mathbf{u}}{\partial \mathbf{t}} + \mathbf{u}\cdot\nabla\mathbf{u}\right) + \nabla\mathbf{p} = \nabla^2\mathbf{u} \;\;,\;\; \nabla\cdot\mathbf{u} = 0 \tag{1}$$

where $Re$ is the Reynolds number of the flow, $\mathbf{u}$ is the (vector-valued) velocity, $\mathbf{p}$ the (scalar-valued) pressure, and $\nabla$ is the gradient. In this equation, $\mathbf{u}\cdot\nabla\mathbf{u}$ is the advection term, $\nabla\mathbf{p}$ is the pressure term and $\nabla^2\mathbf{u}$ is the diffusion term. Key to our approach, the second equation $\nabla\cdot\mathbf{u} = 0$ enforces the conservation of the mass. The difficulty of solving Equation 1 in its non-linear form motivates us to devise a linearization.

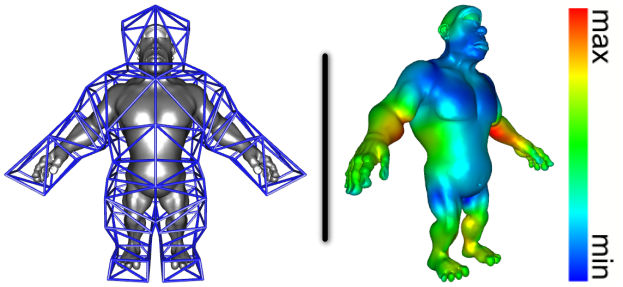


**Figure 4:** ***Cage-based Encoding.*** *As input, we provide the default pose of the* `Ogre` *model with its undeformed shape-aware cage (**left-hand side**). Then, we color-code the root-mean-square error (normalized by the bounding box diagonal length) measuring the distortion resulting of cage-based encoding using our Stokes Coordinates (**right-hand side**).*

**Stokes Linearization** We now derive a linear formulation from the Navier-Stokes equation for our particular rigging problem by restricting the steady flows to be highly viscous and incompressible. Thus, we assume a viscous Newtonian fluid of constant density and constant viscosity with $Re$ small. Then, we linearize Equation 1 by neglecting the unsteady, convective terms and non-linear advection term $\mathbf{u}\cdot\nabla\mathbf{u}$ leading to the following Stokes flow equation:

$$\nabla\mathbf{p} - \nabla^2\mathbf{u} = 0 \;\;,\;\; \nabla\cdot\mathbf{u} = 0 \tag{2}$$

By assuming a flow with zero pressure, we eliminate the pressure components from the Stokes equations:

$$\nabla^2\mathbf{u} = 0 \;\;,\;\; \nabla\cdot\mathbf{u} = 0 \tag{3}$$

**Stokes Biharmonization.** Starting from the linearized Stokes equations (Equation 3), we solve the Newtonian Stokes flow using the vorticity-velocity stream function. To make the solving flow problem tractable in mass continuity, we derive a stream function $\varphi$ for the steady viscous incompressible flow. Since the velocity $\mathbf{u}$ is divergence-free (*i.e.* $\nabla\cdot\mathbf{u} = 0$), we express the stream function $\varphi$ such that $\mathbf{u} = \nabla\times\varphi$. The operator $\times$ denotes the curl as a vector operator describing an infinitesimal rotation. We recall that a vector-valued curl operator is defined from the scalar-valued function. The mass continuity equation specifying the divergence of flow velocity being zero, so we replace the flow velocity by the curl of $\varphi$ in such way that mass continuity is always satisfied: $\nabla^2\mathbf{u} = \nabla^2(\nabla\times\varphi) = 0$. The vorticity $\boldsymbol{\omega}$ of the Stokes flow is defined in terms of its flow velocity by $\boldsymbol{\omega} = \nabla\times\boldsymbol{u}$, so $\boldsymbol{\omega} = \nabla\times(\nabla\times\varphi)$. Assuming $\nabla\cdot\varphi = 0$ the divergence-free property, and $\nabla\times(\nabla\times\varphi) = \nabla(\nabla\cdot\varphi) - \nabla^2\varphi$ a well-known vector calculus identity, then $\nabla^2\varphi = -\nabla\times(\nabla\times\varphi)$.

Defining a stream function $\varphi$ constraints three-dimensional incompressible flow with axisymmetry, while remaining in its 3D form. This implies that the vorticity of the Stokes flow is the negative of the Laplacian of the stream function. Now, we derive the elliptic Stokes stream function from vorticity leading to the following Poisson's equation:

$$\boldsymbol{\omega} = -\nabla^2\varphi \tag{4}$$

connecting the stream function to the vorticity. From physics literature, taking the curl of the Stokes equation and noting that the curl of the gradient of any twice differentiable vector field is zero, the vorticity also satisfies the Laplacian equation $\nabla^2\boldsymbol{\omega} = 0$ and the creeping flow ($Re < 0.1$) imposes the following vorticity stream equation:

$$\nabla^2(\nabla\times\varphi) = \nabla^2\boldsymbol{\omega} \tag{5}$$

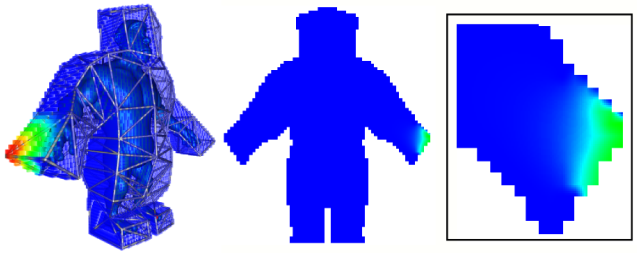

**Figure 5:** ***Stokes Discretization.*** *First, the cage is voxelized and the boundary condition fall-off function is setup for each cage handle (**left-hand side**). Second, the compact second-order Stokes approximation is applied iteratively in the cage interior (**middle and right-hand side**). Finally, we obtain our Stokes Coordinates, at the steady state of the volumetric vorticity stream.*

| Models | | Model #Vert | Model #Face | Cage #Vert | Cage #Face | Model Reduction | Volumetric Grid | $L_2$ max RMSE | $L_2$ min RMSE | $L_2$ avg RMSE |
|---|---|---|---|---|---|---|---|---|---|---|
| Armadillo | (Fig. 5) | 15002 | 30000 | 110 | 216 | 99.3% | $90 \times 90 \times 90$ | 6.9909 | 0.0894 | 2.1562 |
| Horse | (Fig. 5) | 48485 | 96966 | 51 | 98 | 99.8% | $80 \times 80 \times 80$ | 0.2160 | 0.0126 | 0.0752 |
| Old man | (Fig. 1) | 35215 | 702538 | 141 | 278 | 99.5% | $90 \times 90 \times 90$ | 0.2646 | 0.0044 | 0.0757 |
| Ogre | (Fig. 3) | 28352 | 52306 | 98 | 192 | 99.6% | $90 \times 90 \times 90$ | 0.8171 | 0.0153 | 0.0394 |
| Flamingo | (Fig. 7) | 26394 | 52895 | 106 | 208 | 99.5% | $80 \times 80 \times 80$ | 0.0483 | 0.0027 | 0.0141 |
| Camel | (Fig. 7) | 9770 | 19536 | 36 | 68 | 99.6% | $80 \times 80 \times 80$ | 0.0640 | 0.0011 | 0.0018 |
| Hand | (Fig. 7) | 14347 | 28600 | 101 | 198 | 99.2% | $64 \times 64 \times 64$ | 3.4460 | 0.0731 | 1.0265 |

**Table 1:** ***Statistics****: We report a summary of statistics for a various collection of cage-based shapes. We evaluate the distortion injected by our Stokes Coordinates encoding, by measuring the associated Root Mean Square $L_2$ errors (normalized by the bounding box diagonal length) for the identity reproduction and for various cage-based model.*

Finally, the steady state dimension-less vorticity becomes:

$$\nabla^2 \mathbf{u} = \nabla^2 \boldsymbol{\omega} = \nabla^2 \left( -\nabla^2 \boldsymbol{\varphi} \right) = 0 \tag{6}$$

In steady state, we obtain the Stokes-wise biharmonic equation for the stream function, where the components of $\mathbf{u}$ solves the biharmonic limit of the Stokes equations [Mohanty et al. 2011]. Consequently,

$$\nabla^2 \left( \nabla^2 \boldsymbol{\varphi} \right) = \nabla^4 \boldsymbol{\varphi} = 0 \tag{7}$$

where $\boldsymbol{\varphi}$ is the scalar-valued biharmonic function, $\nabla^4$ is the fourth power of the Del operator (also known as biharmonic operator), subject to Dirichlet boundary conditions in the form of a fall-off function for each cage handle. $\nabla^2$ is the square of the Laplacian operator. In 3D Cartesian coordinates the Stokes-wise biharmonic equation is a fourth-order partial differential equation is written as:

$$\nabla^4 \boldsymbol{\varphi}(x,y,z) = \frac{\partial^4 \boldsymbol{\varphi}}{\partial x^4} + \frac{\partial^4 \boldsymbol{\varphi}}{\partial y^4} + \frac{\partial^4 \boldsymbol{\varphi}}{\partial z^4} + 2\frac{\partial^4 \boldsymbol{\varphi}}{\partial x^2 \partial y^2} + 2\frac{\partial^4 \boldsymbol{\varphi}}{\partial x^2 \partial z^2} + 2\frac{\partial^4 \boldsymbol{\varphi}}{\partial y^2 \partial z^2}$$

with $(x,y,z) \in \Omega$, defined as the inner space bounded by the polyhedral cage.

**Compact Second-Order Operator.** Using the second-order approximation suggested by Altas *et al.* [Altas et al. 2002], we construct the unique solution of the Stokes equations as a scalar-valued biharmonic function $\boldsymbol{\varphi}(x,y,z)$ satisfying Equation 7. For high numerical accuracy, we discretize the approximation with a 18-point finite centered differences. The compact stencil is defined at each grid corner point as an approximation scheme limited to two-rings compact cell. Considering a 3D uniform grid centered at the point $\left(x_i, y_j, z_k\right)$ values of the desired solution $\boldsymbol{\varphi}$, the compact finite-difference approximations of the Stokes biharmonic operator is expressed as follows

$$\begin{aligned} q_{i,j,k} = \frac{1}{48} [\ & 10\left(q_{i+1,j,k} + q_{i,j+1,k} + q_{i,j,k+1}\right) \\ & + 10\left(q_{i-1,j,k} + q_{i,j-1,k} + q_{i,j,k-1}\right) \\ & - q_{i+1,j,k+1} - q_{i,j+1,k+1} - q_{i-1,j,k+1} - q_{i,j-1,k+1} \\ & - q_{i+1,j,k-1} - q_{i,j+1,k-1} - q_{i-1,j,k-1} - q_{i,j-1,k-1} \\ & - q_{i+1,j+1,k} - q_{i-1,j+1,k} - q_{i-1,j-1,k} - q_{i+1,j-1,k} \\ & - 3h\left(q_{x_{i+1,j,k}} - q_{x_{i-1,j,k}} + q_{y_{i,j+1,k}} - q_{y_{i,j-1,k}} + q_{z_{i,j,k+1}} - q_{z_{i,j,k-1}}\right) ] \\ & - \frac{h^4}{2}\boldsymbol{\varphi}(i,j,k) \end{aligned} \tag{8}$$

The scalar value at the point at the grid point $(i,j,k)$ are written $q_{i,j,k}$. The formulation also incorporates the finite-difference approximations gradients functions written as $q_x, q_y, q_z$ at the point $(i,j,k)$. We refer the interested reader to the excellent paper of Altas *et al.* [Altas et al. 2002] for further mathematical details about this operator.

**Volumetric Stokes Weights Solver.** We perform iterative weights conduction our *Stokes Coordinates* using a volumetric solver. To make the implementation amenable, we discretize the Stokes operator inside the voxelized domain of the cage. Since the approximations use the 18-point compact stencil with a two-rings grid cells neighborhood (Equation 7), Dirichlet boundary conditions are incorporated similarly to [Joshi et al. 2007]. Figure 5 shows the grid discretization employed as the support for the Stokes-wise biharmonic operator. For each cage handle, we fix a specific boundary condition and then we apply the Stokes-wise biharmonic operator (Equation 8) iteratively on the grid cell in the cage interior. The propagation of boundary condition values mimics the fluid flow using several Jacobi-like iterations that iteratively update the weight map. Figure 5 displays a cut-away slice of the voxelized cage interior and a close-up visualization of the cage-handle boundary condition transport acting as fluid-weights propagated through the grid. Our cage-voxelized finite difference method is similar in spirit to [Joshi et al. 2007] but enriched with a compact second-order operator. The optimal solution for $\boldsymbol{\varphi}$ is reached at convergence. The iteration termination criteria is reached when all per-voxel values variations between two iterations lies under a given threshold, assumed to be sufficient enough for shape deformation applications. Then, *Stokes Coordinates* are extracted via trilinear interpolation of the grid cell values $q_{i,j,k}$ (Equation 8). To ensure partition of unity, we rescale resulting coordinates to ensure that they sum up to 1. For each model vertex $j$, we defined a *Stokes Coordinates* weighting function $\boldsymbol{w}_j(l) \in \mathbb{R}$ where $l$ is a given cage handle. To achieve cage-based deformation, *Stokes Coordinates* serve as weights to compute the new location of deformed model vertices $\boldsymbol{p}'_j$ as follows, assuming $\boldsymbol{c}'_l$ is the deformed location of the cage handle $l$:

$$\boldsymbol{p}'_j = \sum_l \boldsymbol{w}_j(l) \cdot \boldsymbol{c}'_l \quad \text{subject to} \quad \forall l, \quad \sum_j \boldsymbol{w}_j(l) = 1 \tag{9}$$

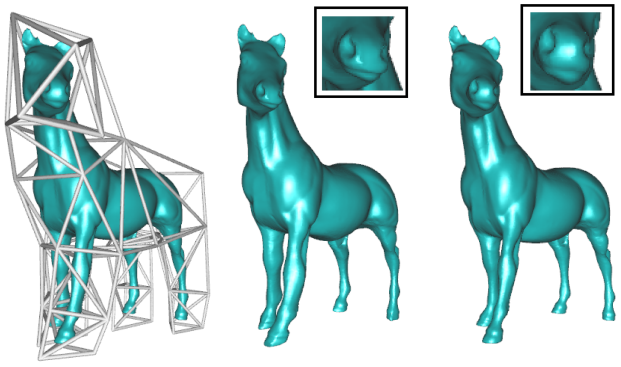

**Figure 6:** ***Cage-based Shapes.*** *The* `Horse` *model is enclosed in a coarse cage (**left**). We compare side-by-side the cage-based shape with mean value coordinates (**middle**), and the cage-based shape with Stokes Coordinates (**right**).*

## 4 Experimental Results and Evaluation

We have implemented a standalone prototype of our compact *Stokes Coordinates* in C/C++. We tested our algorithm using an Alienware workstation with 4Gb memory and a Geforce GTX 660 graphics card. Our framework offers three cage-based applications employing our novel coordinates system: interactive *cage-based mesh editing*, *cage-based shape encoding*, and *cage-based performance capture reuse*.

**Cage-based Editing and Animation.** We perform interactive cage based editing using our *Stokes Coordinates*. The cage can be deformed using Laplacian, *As-Rigid-As Possible* or linear interpolation techniques. As shown in Figure 3, our coordinates system is used for data-driven shape deformation to output collection of animated poses for the `Horse` and the `Armadillo` model.

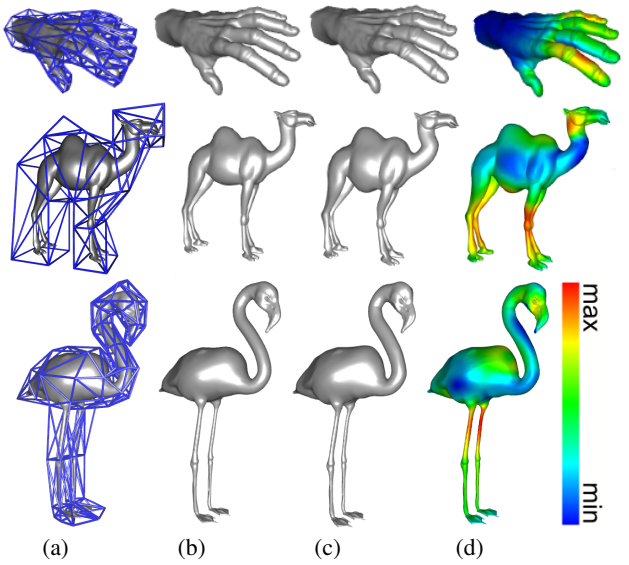


**Figure 7:** ***Cage-based Encoding.*** *We display the default input model surrounding by its corresponding coarse cage (**a**), the default input model (**b**) and the cage-based model encoded with our Stokes Coordinates (**c**). Then, we color-code the root-mean-square error (normalized by the bounding box diagonal length) measuring the distortion injected by our Stokes Coordinates cage-based encoding (**right-hand side**), for the* `Hand`*,* `Camel`*,* `Flamingo` *model (**from top to bottom row**).*

**Cage-based Shape Encoding.** A well-known problem with cage-based encoding is the conformality. A default cage with our coordinates can be interpreted as a re-skinning mesh technique that is not restricted to quasi-articulated or piece-wise rigid shapes. Figure 6 illustrates a comparison of *Mean Value Coordinates* and *Stokes Coordinates*. Our coordinates finely encode the lips and jaws of the `Horse` with quasi-conformal properties. Even if the cage is a global underlying structure, our *Stokes Coordinates* can be served as local shape descriptor. Various cage-based encoding for complex cage-based organic shapes are shown at low-distortion rate. Reconstructing the enclosed surface with the default case allows us to compare side-by-side with the default model. Figures 4 and 7 show the measured encoding RMS errors. Table 1 reports performance statistics. The error resulting of the *Stokes Coordinates* are somehow correlated with the design choice for the default cage. This correlation makes the accuracy of *Stokes Coordinates* highly dependent on the cage tessellation and thus hard to evaluate in itself. We point out that it is challenging for artists to design a cage for complex organic topologies, even at the rest pose. Nevertheless, local fine details are acceptably preserved if the cage is sufficiently shape-aware.

**Cage-based Performance Reuse.** The reuse of captured dynamic mesh is now a crucial problem at the center of various domains of application like filmmaking or cartoon animation production for real-world captured data. We reuse performance capture meshes by decomposing the template shape into a collection of cage handles. Also, squash and stretch is a fundamental principle in animation. The right-hand side of Figure 1 depicts cage-driven surface stretching. We show the feasibility of reusing performance capture meshes by decomposing the static template shape into a collection of cage handles coupled with our *Stokes Coordinates*. The humanoid body is locally squashed and stretched by pushing and pulling cages vertices. Our *Stokes Coordinates* allow non-isometric deformations necessary to achieve cartoon-style over life-like shapes.

## 5 Discussion

In this section, we propose a discussion of our approach devised for fluid-inspired coordinates and its implementation, as well as our results obtained with the proposed techniques.

**Fluid-Inspired Coordinates.** The fluid principle brings the intuition about how to assign high-order weights to a model for the cage-based rigging problem. In particular, vorticity transport offers a practical linear numerical framework to solve cage-based weighting functions by propagating weight following a fluid-flow fashion. Our method required a straightforward voxelization procedure with the finite difference to compute high-order coordinates. Finally, our coordinates system allows rich geometry variations, non-rigid deformations and expressive control of the enclosed surface with few user interactions. Independently, our work also provides a theoretical understanding of harmonicity property for cage coordinates via the general flows analogy. Table 2 proposes a high-level comparison between our *Stokes Coordinates* with previous approaches.

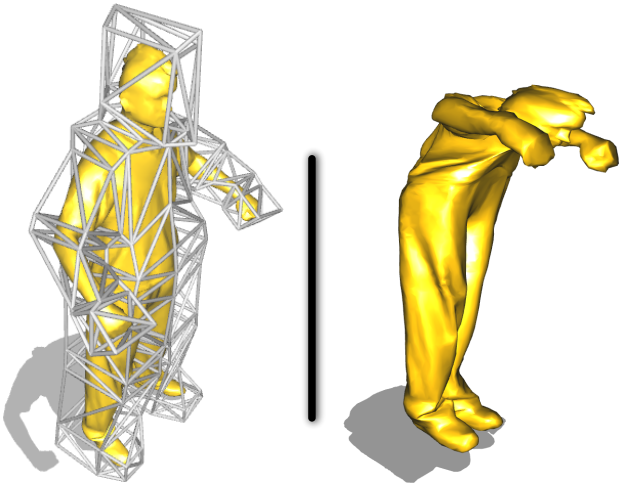

**Figure 8:** ***Squash and Stretch Deformation.*** *A humanoid model obtained by performance capture is enclosed inside a bounding cage (**left-hand side**). We produce a cage-based shape with local non-rigid deformation reproducing cartoon-style squash and stretch deformation (**right-hand side**). Using our proposed coordinates, the resulting non-isometric shape deformation is fluid-like looking with local volume preserving.*

**Linear Cage-Aware Stokes Flow.** The principal limitation of the Stokes linearization is the restriction of the fluid vorticity, limiting the full potential of fluid dynamic for coordinates computation. Even if Equation 7 is well-known and has already been applied to solving cage coordinates, our derivation is new and pushes forward the literature of biharmonic cage coordinates by expanding deformation to the level of the flow. Tying bi-harmonic functions to stokes flows could offer fine tailoring of the weights transportation by controlling the flow velocity and the curl as the circulation density of the weight. Also, this sacrifice does not damage the mass-conserving property of resulting coordinates, allowing rich and expressive non-rigid control with a shape-aware cage. Our unoptimized implementation suffers from computational cost and memory overhead since the solver converge of the solver depends on the resolution of the discrete uniform voxel grid. This intermediate discontinuous discretization introduce undesired artifacts. This issue can be solved by multi-grid optimization.

The results demonstrate that cage-based weights lead to satisfactory control for surface deformation. The iterative scheme to obtain the Stokes flow solution is very effective with highest strain accuracy. However, starting from a viscous flow for coordinates computation does not implies that resulting surface deformations produced by estimated coordinates will depict physically-valid fluid effects. In addition, solutions of the linearized Stokes equation do not satisfy the maximum principle property. The missing reproduction property and the low degree of smoothness of the Stokes Coordinates is inherent to all other coordinates system since conformality in 3D seems to be impossible. Bounding the generated weight functions may lead to better intuitive responses.

| Properties *Coordinates* | HC | MVC | BHC | GC | Ours |
|---|---|---|---|---|---|
| Closeness | no | yes | no | yes | no |
| Finite Difference | yes | no | - | no | yes |
| Stretchability | yes | yes | yes | no | yes |
| High-order | no | no | yes | no | yes |
| Stencil Compactness | - | - | no | - | yes |
| Local Incompressibility | no | no | no | no | yes |

**Table 2:** ***High-level Comparison.*** *We present a comparison table of our Stokes Coordinates with previous approaches: Harmonic Coordinates (HC), Mean Value Coordinates (MVC), Bi-Harmonic Coordinates (BHC), Green Coordinates (GC).*

**Compact-Stenciled Vorticity Transport.** The proposed high-order coordinates are resulting from the solution of the vorticity stream equation for the Stokes flow at the steady state. Our *Stokes Coordinates* encode shape representation at low-distortion rate while preserving mass and density. Linearization and biharmonization are two key steps to discretize the Stokes flow over the volumetric domain defined by the low-dimensional cage structure. Since our flow is not assumed to be irrotational ($\nabla \times u \neq 0$), then $\nabla^2\varphi = 0$ is not a trivial solution for the solved system. We demonstrate the feasibility of deriving biharmonic weights for cage-based deformation from Stokes's transport equation. Our second-order compact solver only requires a compact stencil while retaining high-order accurate approximation properties of its classical version. This technique remains center-differencing in spirit. Finally, the benefit of fluid-perspective for geometry deformation is that viscosity parameters could potentially offer better control. Still, the compactness of the employed stencil does not implies the compactness of the coordinates. Then, the compactness only refers to the stencil and not the result weights.

**Local Incompressibility.** The volume-preserving property of the virtual fluid is obtained thanks to the solenoidal divergence-free vector field $\nabla \cdot \mathbf{u} = 0$. However, there is not guarantee that the global volume of the deformed mesh with Stokes coordinates is preserved. For stretchable cage-based deformation, the volume conservation is highly subject to hard constraints expressed by how the cage deformed. Then, global incompressibility is not an issue we need to pay serious attention. However, it is reasonable that Stokes coordinates bring the local incompressibility into the cage-based deformation. Since the object should also shrink when the cage shrinks, Equation 9 gives control over the change in volume upon global deformation by construction. Our incompressibility formulation only conserves local volumes over isometric sub-region while allowing *as non-isometric as possible* constrained deformation. In our work, we demonstrate that non-rigid isometric deformations have a vorticity propagation behavior reflecting the fluid flow. We estimate the optimal stretch-minimizing stream since non-isometric stretching energy is minimized by solving an incompressible Stokes equation using the finite difference method discretized over the voxelized domain.

## 6 Conclusions and Future Work

In this paper, we describe a method for generating mesh bind weights for control cages built upon the conceptual fluid-cage connection. Inspired by the incompressible flow theory, we introduced a new high-order coordinates system for cage-based rigging called *Stokes Coordinates*, allowing non-isometric stretching. We borrow the idea from fluid dynamics to express shape deformation as a Stokes problem. In particular, we explored stencil computation and governing field equations for cage weighting functions.

**Stokes Cage Coordinates.** In this paper, we take inspiration from incompressible fluid transport to derive weight functions for cage-based deformation. Since the Stokes flow is difficult to extend in 3D, we have developed a Stokes rigging solver by taking benefit from a linearization and biharmonization formulation. In the context of deformable cages, Stokes flow serves as cage weighting functions while providing useful incompressibility and mass-conserving properties. More importantly, we compute high-order cage coordinates efficiently by relying on the Stokes stream function at steady state. We have developed a stand-alone implementation for the accurate compact scheme. The equation is discretized on a regular voxel grid and solved in Jacobi-style fashion by repeated application of a finite-difference stencil. Our method does not corrupt the classical cage metaphor, nor does it modify the deformation equations. Furthermore, we show that our *Stokes Coordinates* can be volumetrically discretized and approximated using only a two-rings grid cells stencil. Further, we would like to investigate how to pilot the vorticity (for fluid dynamic) and stream function equation (for fluid kinematic) to allow controllable weight transport. Applying the Stokes flow principle to cage problems is interesting since our new formulation brings incompressibility and localized volume-preserving properties to cage-based deformation allowing non-isometric stretching.

**Future Work.** Representing shape deformation as incompressible flow is a very interesting solution to the cage-based rigging problem allowing volume control. We proposed a vorticity stream strategy dedicated to high-order coordinates for free-form cages. We believe our alternative opens new directions for interactive shape modeling. In the future, we hope to improve the convergence rate of our basic iterative method. A natural extension is to consider fast rebinding for shape deformation involving topological changes. Our new fluid-inspired coordinates system could be extended to skeleton-based deformation or volumetric diffusion curves. Our new coordinate system could benefit to geometric re-

verse engineering like animation re-skinning or shape decomposition. We are optimistic that our technique could be useful for volumetric diffusion curves or field-guided registration driven by a small set of cage-handles. Straightforwardly, a potential future work is to rely on an out-of-core multigrid solver with GPU parallelization or block-SOR iteration. Another interesting avenue is to generalize our compact discretization to any irregular grid. Finally, we want to extend our compact-stenciled strategy to triharmonic or quadriharmonic weighting functions. More importantly, we would like to put efforts in investigating if our method could be of benefit to animating creatures made of liquid, like amoeba, slugs and so forth.

## References


ALTAS, I., ERHEL, J., AND GUPTA, M. M. 2002. High accuracy solution of three-dimensional biharmonic equations. *Numerical Algorithms*.

ANDO, R., THUEREY, N., AND WOJTAN, C. 2015. A stream function solver for liquid simulations. *ACM Trans. Graph*.

ANGELIDIS, A., AND SINGH, K. 2007. Kinodynamic skinning using volume-preserving deformations. In *SCA*.

BARAN, I., AND POPOVIĆ, J. 2007. Automatic rigging and animation of 3d characters. *ACM Trans. Graph*.

BARBIČ, J., AND JAMES, D. L. 2005. Real-time subspace integration for st. venant kirchhoff deformable models. *ACM Trans. Graph*.

BEN-CHEN, M., WEBER, O., AND GOTSMAN, C. 2009. Variational harmonic maps for space deformation. *ACM Trans. Graph*.

BUDNINSKIY, M., LIU, B., TONG, Y., AND DESBRUN, M. 2016. Power coordinates: A geometric construction of barycentric coordinates on convex polytopes. *ACM Trans. Graph.*.

CAI, Z., WANG, C., AND ZHANG, S. 2010. Mixed finite element methods for incompressible flow: Stationary navier-stokes equations. *SIAM J. Numer. Anal*.

CAPELL, S., BURKHART, M., CURLESS, B., DUCHAMP, T., AND POPOVIĆ, Z. 2005. Physically based rigging for deformable characters. In *SCA*.

CHILDRESS, S., 2007. Fluid dynamics: Stokes flow (chapter 7).

FENG, P., AND WARREN, J. 2012. Discrete bi-laplacians and biharmonic b-splines. *ACM Trans. Graph*.

FISHELOV, D., BEN-ARTZI, M., AND CROISILLE, J.-P. 2011. Highly accurate discretization of the navier stokes equations in streamfunction formulation. In *Spectral and High Order Methods for Partial Differential Equations*.

GARCÍA, F. G., PARADINAS, T., COLL, N., AND PATOW, G. 2013. *cages:: A multilevel, multi-cage-based system for mesh deformation. *ACM Trans. Graph*.

GREENSPAN, D., AND SCHULTZ, D. 1972. Fast finite-difference solution of biharmonic problems. *Commun. ACM*.

HAHN, F., MARTIN, S., THOMASZEWSKI, B., SUMNER, R., COROS, S., AND GROSS, M. 2012. Rig-space physics. *ACM Trans. Graph*.

JACOBSON, A., BARAN, I., POPOVIĆ, J., AND SORKINE, O. 2011. Bounded biharmonic weights for real-time deformation. In *ACM SIGGRAPH*.

JACOBSON, A., WEINKAUF, T., AND SORKINE, O. 2012. Smooth shape-aware functions with controlled extrema. *Comp. Graph. Forum*.

JACOBSON, A., WEINKAUF, T., AND SORKINE, O. 2012. Smooth shape-aware functions with controlled extrema. *SGP*.

JOSHI, P., MEYER, M., DEROSE, T., GREEN, B., AND SANOCKI, T. 2007. Harmonic coordinates for character articulation. *ACM Trans. Graph*.

JOVANOVIC, M., AND BAMIEH, B. 2001. Modeling flow statistics using the linearized navier-stokes equations. In *Decision and Control*.

JU, T., SCHAEFER, S., AND WARREN, J. 2005. Mean value coordinates for closed triangular meshes. In *ACM SIGGRAPH*.

JU, T., ZHOU, Q.-Y., VAN DE PANNE, M., COHEN-OR, D., AND NEUMANN, U. 2008. Reusable skinning templates using cage-based deformations. In *ACM SIGGRAPH*.

KAROLY, Z., AND LASZLO, S.-K. 2013. Real-time control of newtonian fuids using the navier-stokes equations. *KEPAF*.

KAVAN, L., AND SORKINE, O. 2012. Elasticity-inspired deformers for character articulation. *ACM Trans. Graph*.

KIM, T., AND JAMES, D. L. 2011. Physics-based character skinning using multi-domain subspace deformations. In *SCA*.

LE, B. H., AND DENG, Z. 2017. Interactive cage generation for mesh deformation. In *I3D*.

LI, X.-Y., JU, T., AND HU, S.-M. 2013. Cubic mean value coordinates. *ACM Trans. Graph*.

LIPMAN, Y., LEVIN, D., AND COHEN-OR, D. 2008. Green coordinates. In *ACM SIGGRAPH*.

MCADAMS, A., ZHU, Y., SELLE, A., EMPEY, M., TAMSTORF, R., TERAN, J., AND SIFAKIS, E. 2011. Efficient elasticity for character skinning with contact and collisions. In *ACM SIGGRAPH*.

MOHANTY, R. K., JAIN, M. K., AND MISHRA, B. N. 2011. A new fourth order difference approximation for the solution of three-dimensional non-linear biharmonic equations using coupled approach. *AJCM*.

RANNACHER, R. 2000. Finite element methods for the incompressible navier-stokes equations. In *Fundamental Directions in Mathematical Fluid Mechanics*.

SEGAL, A., UR REHMAN, M., AND VUIK, C. 2010. Preconditioners for incompressible navier-stokes solvers. *Numerical Mathematics: Theory, Methods and Applications*.

SHANG, Y., HE, Y., WAN KIM, D., AND ZHOU, X. 2011. A new parallel finite element algorithm for the stationary navier stokes equations. *Finite Elem. Anal. Des*.

SHANG, Y. 2013. A two-level subgrid stabilized oseen iterative method for the steady navier stokes equations. *J. Comput. Phys*.

SHEN, C., ZHANG, G., LAI, Y., HU, S., AND MARTIN, R. 2010. Harmonic field based volume model construction from triangle soup. *Journal of Computer Science and Technology*.

TURNER, J. A., AND MAZZONE, A. C. 1999. Multifluid finite volume navier stokes solutions for realistic fluid animation. In *ACM SIGGRAPH (abstracts and applications)*.

UR REHMAN, M., VUIK, C., AND SEGAL, G. 2010. Block preconditioners for the incompressible stokes problem. In *Large-Scale Scientific Computing*.

VON FUNCK, W., THEISEL, H., AND SEIDEL, H.-P. 2007. Elastic secondary deformations by vector field integration. In *SGP*.

WANG, H., SIDOROV, K. A., SANDILANDS, P., AND KOMURA, T. 2013. Harmonic parameterization by electrostatics. *ACM Trans. Graph.*.

WANG, Y., JACOBSON, A., BARBIČ, J., AND KAVAN, L. 2015. Linear subspace design for real-time shape deformation. *ACM Transactions on Graphics (TOG)*.

WEBER, O., PORANNE, R., AND GOTSMAN, C. 2012. Biharmonic coordinates. *Comput. Graph. Forum*.

XU, H., AND HE, Y. 2013. Some iterative finite element methods for steady navier stokes equations with different viscosities. *J. Comput. Phys*.

XU, K., ZHANG, H., COHEN-OR, D., AND XIONG, Y. 2009. Dynamic harmonic fields for surface processing. *Computers and Graphics (Special Issue of Shape Modeling International)*.

ZHANG, J., DENG, B., LIU, Z., PATANÈ, G., BOUAZIZ, S., HORMANN, K., AND LIU, L. 2014. Local barycentric coordinates. *ACM Trans. Graph*.

ZHENG, C., AND JAMES, D. L. 2009. Harmonic fluids. *ACM Trans. Graph*.