

Filament Surgery and Temporal Grid Adaptivity Extensions to a Parallel Tree Code for Simulation and Diagnosis in 3D Vortex Dynamics

Victor M. Fernandez, Norman J. Zabusky
Department of Mechanical and Aerospace Engineering and CAIP Center
Rutgers University
Piscataway, NJ 08855, USA

Pangfeng Liu^{1,2}, Sandeep Bhatt^{1,2}, Apostolos Gerasoulis¹

¹Bell Communications Research
Morristown, NJ 07960, USA

²Department of Computer Science
Rutgers University, Piscataway, NJ 08855, USA

Abstract

The vortex collapse-reconnection process presents behaviors commonly observed in turbulent flows: multiple spatial and temporal scales, rapid vorticity and strain-rate amplification and dissipation through generation of small scales. In this work we reduce the computational complexity of our problem by using hierarchical methods (tree codes), introducing a time extrapolation framework for each particle, and applying a filament “surgery” algorithm, based on the energy density, as a regularization for vortex collapse. The high performance parallel implementation of Barnes-Hut algorithm permit us to increase by one order of magnitude the resolution of the vortex collapse simulations. The use of the time extrapolation for slow moving particles helps in concentrating the computational effort in the important dynamic domains. The vortex filament surgery regularizes effectively the growth of the number of particles in the collapse regions of the flow. The reduction in complexity achieved will contribute to optimize the use of the numerical simulations in the reduced model building process.

1 Introduction

The study of small scales formation in high Reynolds number flows is of fundamental importance because of the relationship this process has with important areas of engineering applications, including drag reduction and mixing [17]. The study of these types of flows tends to require substantial amount of computational resources. In multi-filament models, the $O(n^2)$ nature of the computational expense of the Biot-Savart direct method (where n is the number of grid points) severely limits the resolution of the simulations [10]. The objective of our work is to adapt a high performance $O(n \log n)$ parallel implementation of N-Body algorithms [6, 16] to the vortex dynamics problem and extend it by including a filament surgery algorithm based on the energy density [11]. We also exploit the multiple time scales in the flow to concentrate the computational effort in regions with higher dynamic activity. We emphasize the use of the fast summation techniques for the computation of diagnostics.

2 A tree code for vortex dynamics

The version of the vortex method we use was developed by Knio and Ghoniem [13]. The vorticity of a vortex tube is represented by a bundle of vortex filaments $\chi_i(\sigma, t^*)$, each of them with circulation Γ_j

$$\omega(\mathbf{x}, t^*) = \sum_{j=1}^{N_f} \Gamma_j \int_C d\chi(\sigma_j, t^*) f_\delta(|\mathbf{x} - \chi(\sigma_j, t^*)|) . \tag{1}$$

The total number of filaments forming the vortex tube is N_f and σ_j is the initial arc length along each filament. The core function f_δ is given by the third-order Gaussian

$$f_\delta(\rho) = \frac{3}{4\pi\delta^3(\sigma, t^*)} \exp(-\rho^3/\delta^3(\sigma, t^*)) , \tag{2}$$

where δ is the core “radius”.

The ordering of the grid points in the filaments is only important when computing the differentials

$$d\chi \sim \Delta\chi_j = \frac{1}{2} (\chi_{j+1} - \chi_{j-1}) . \tag{3}$$

This is a characteristic of the filament approach in 3D vortex methods. In contrast with the “vortex arrow” algorithm [15, 22], updating the “strength” of the vortex elements in the filament method does not require the evaluation of the velocity gradient, which involves the computation of an additional integral over all of the particles. Also, filaments with form of closed curves, satisfy the divergence free condition of the vorticity field. This is not always the case at all times in the vortex arrow approach.

After computing the differentials the order of the summation is unimportant. Therefore the vorticity distribution may be discretized by

$$\omega(\mathbf{x}, t^*) = \sum_{p=1}^{N_p} \alpha_p f_\delta(|\mathbf{x} - \chi_p(t^*)|) . \tag{4}$$

where

$$\alpha_p = \Gamma_p \Delta\chi_p . \tag{5}$$

With this vorticity distribution the stream function becomes

$$\Psi(\mathbf{x}) = \sum_{p=1}^{N_p} \alpha_p \psi(|\mathbf{x} - \chi_p|) , \tag{6}$$

and the velocity field is

$$\mathbf{u}(\mathbf{x}) = - \sum_{p=1}^{N_p} \alpha_p \times \nabla \psi(|\mathbf{x} - \chi_p|) , \tag{7}$$

where

$$\psi(\rho) = - \int_0^\rho \frac{ds}{s^2} \int_0^s f_\delta(t) t^2 dt + \psi(0) . \tag{8}$$

We use the dimensionless time $t = t^*/t_s = t^* \Gamma_C / (4\pi a^2)$, where Γ_C and a are defined in 19.

A fast evaluation of the stream function Ψ and the velocity field \mathbf{u} can be obtained by using the Taylor's expansion of the function ψ . Using the notation by Band [3] and Schwartz [23] for the powers $((\boldsymbol{\chi} - \boldsymbol{\chi}_0) \cdot \nabla)^n$, we obtain the multi-pole expansion of the stream function

$$\Psi(\mathbf{x}) = \sum_{n=0, j, k}^{m, j+k \leq n} \mathbf{M}(j, k, n - j - k) D(j, k, n - j - k) \psi(\mathbf{x} - \boldsymbol{\chi}_0) + \Phi_m \quad (9)$$

and the multi-pole expansion of the velocity field

$$\mathbf{u}(\mathbf{x}) = - \sum_{n=0, j, k}^{m, j+k \leq n} \mathbf{M}(j, k, n - j - k) \times D(j, k, n - j - k) \nabla \psi(\mathbf{x} - \boldsymbol{\chi}_0) + \nabla \times \Phi_m, \quad (10)$$

where

$$\mathbf{M}(j, k, n - j - k) = \sum_{p=1}^{N_p} \boldsymbol{\alpha}_p (\chi_p - \chi_0)_1^j (\chi_p - \chi_0)_2^k (\chi_p - \chi_0)_3^{n-j-k}, \quad (11)$$

$$D(j, k, n - j - k) = \frac{(-1)^n}{(n - j - k)! j! k!} \partial_1^j \partial_2^k \partial_3^{n-j-k} \quad (12)$$

and

$$\Phi_m = \sum_{p=1}^{N_p} \boldsymbol{\alpha}_p K_m, \quad (13)$$

with

$$K_m(\mathbf{x}, \boldsymbol{\chi}_p) = \sum_{j, k}^{j+k \leq m+1} \frac{(m+1)! (\chi_p - \chi_0)_1^j (\chi_p - \chi_0)_2^k (\chi_p - \chi_0)_3^{m+1-j-k}}{(m+1-j-k)! j! k!} \partial_1^j \partial_2^k \partial_3^{m+1-j-k} \psi(\mathbf{x} - \boldsymbol{\chi}_0 - t(\boldsymbol{\chi}_p - \boldsymbol{\chi}_0)), \quad (14)$$

and $0 \leq t \leq 1$.

As a diagnostics for vortex collapse, we compute the strain-rate and also the energy density, which can be evaluated fast by using the multi-pole expansion of Ψ .

The error of the truncated multi-pole expansion has been estimated for the gravitational problem, among others, by Salmon and Warren [20, 21]. We have checked that their estimate applies to the multi-pole expansion in the vortex dynamics problem. For the point vortex Green's function $\psi = 1/(4\pi\rho)$, the weak bounds [21] of the error are, for the vector stream function

$$|\Phi_m(\mathbf{x})| \leq C_0 \left(\frac{b}{d}\right)^{m+1}, \quad (15)$$

and for the velocity field

$$|\nabla \times \Phi_m(\mathbf{x})| \leq C_1 \left(\frac{b}{d}\right)^{m+1}. \quad (16)$$

The constants are

$$C_0 = \frac{1}{d} \frac{1}{1 - b/d} \sum_{p=1}^{N_p} |\boldsymbol{\alpha}_p|, \quad (17)$$

and

$$C_1 = \frac{1}{d^2} \frac{1}{(1 - (b/d)^2)} (m + 2 - \frac{b}{d}(m + 1)) \sum_{p=1}^{N_p} |\boldsymbol{\alpha}_p|, \quad (18)$$

where $b = \max |\chi_p - \chi_0|$ is the size of the cluster and $d = |\mathbf{x} - \chi_0|$ is the distance to the cluster. More precise estimates are also available by Salmon and Warren [22] and more recently by Winckelmans et al. [25] for the vortex dynamics problem.

To reduce the computational complexity in the sums in Eq. 10, we use [6] the hierarchical method by Barnes and Hut [4] for gravitational fields. To organize a hierarchy of clusters, we first compute an oct-tree partition of the three-dimensional box enclosing the vortex elements. Once the oct-tree has been built, the moments of the internal nodes are computed in one phase up the tree, starting at the leaves. The next step is to compute induced velocities; each particle traverses the tree in depth-first manner starting at the root. For any internal node, if the distance D from the corresponding box to the particle exceeds the quantity R/θ , where R is the side-length of the box and θ is an accuracy parameter, then the effect of the subtree on the particle is approximated by a particle-to-cluster interaction, where the cluster is represented by point vortex multi-poles located at the geometric center of the tree node. The tree traversal continues, but the subtree is bypassed. Once the induced velocities of all the bodies are known, the new positions and vortex element strengths are computed. The entire process, starting with the construction of the oct-tree, is repeated for the desired number of time steps. For convenience we refer to the set of nodes which contribute to the velocity on a particle as the *essential* nodes for the particle [6]. Each particle has a distinct set of essential nodes which changes with time. One remark concerning distance measurements is in order. There are several ways to measure the distance between a particle and a box. Salmon [20] discusses various alternatives in some detail. For consistency, we measure distances from bodies to the perimeter of a box in the L_∞ metric. In our experiments we vary θ to test the accuracy of the code compared to solutions computed using the direct method at specific points. The tree code has been implemented on a CM5 with SPARC vector units. The simulation sustains an overall rate of over 37 Mflop / sec per node [6].

Our initial condition is a multi-filament vortex ring constructed around the center line

$$(\chi_1, \chi_2, \chi_3) = (a \cos \theta, b \sin \theta, c \sin 2\theta), \quad (19)$$

where $0 \leq \theta \leq 2\pi$ and which we call the “Lissajous-elliptic” ring [10]. The thickness of the multi-filament ring is δ_C . The circulation distribution Γ_p , and the initial filament core radius δ_p also need to be specified. Low aspect ratio elliptic rings $a > b, c = 0$ have periodic behavior that can be used for dynamic testing. The initial testing of the code has been carried out with two cases. The first is a low aspect ratio elliptic ring with $a = 1, b = 0.8$ and $c = 0$. The second case tested corresponds to a collapsing vortex ring with $a = 1, b = 0.4$ and $c = 0.5$. The collapse state is shown in Fig. 2. This ring has $N_f = 312$ filaments with a total number of particles $N_p = 319\,488$.

3 Multiple time scales and temporal grid adaptivity

It has been natural in astrophysics to recognize and exploit the hierarchical structures on different time scales that occur in gravitational systems. In 1963, Aarseth [1] introduced the use of variable and individual time step for each particle (star). Some time later (1973), Ahmad and Cohen [2] linked this idea to a hierarchy in space. They separated force contributions in a slow varying force due to distant stars (*regular component*) and a highly fluctuating force induced by immediate neighbors (*irregular component*). More recently, temporal extrapolations have been introduced in the multi-pole expansion by Jernigan and Porter [12], McMillan and Aarseth [18] and Sundaram [24].

The immediate options to exploit different time scales are either to extrapolate force at slow moving particles or extrapolate contribution by distant particles (extrapolation of multi-pole mo-

ments, i.e. temporal multi-pole expansion). An optimal scheme is obtained for well separated time scales (like in turbulent flows). We started considering these ideas by exploring the first approach.

We keep the original time framework of our previous algorithm, which we call now the global time. The extrapolation of the velocity field for the slow moving particles is carried out by introducing a local particle time reference system (Fig. 1). In the global time framework, the velocity is evaluated using the full Biot-Savart integral. The second order predictor-corrector scheme requires two velocity evaluations. Using these, plus one evaluation from the previous time step (three evaluations in total), we construct a cubic polynomial extrapolation for the position in the local particle time

$$\chi((k+1)h) = \chi(0) + (k+1)h \tilde{u}(0) + \frac{1}{2}(k+1)^2 h^2 \tilde{u}'(0) + \frac{1}{6}(k+1)^3 h^3 \tilde{u}''(0) + d_{k+1}, \quad (20)$$

where \tilde{u}' and \tilde{u}'' are central differences obtained using u_{n-1} , u_n and u_{n+1} in the global time frame (χ, u represent position and velocity components). This extrapolation for the particle position is equivalent to a third order Runge-Kutta integration scheme with velocity evaluated using a quadratic extrapolation.

The local truncation error in the extrapolation

$$d_{k+1} = \frac{1}{24} ((k+1)^4 - 2(k+1)^2) h^4 u'''(0) + O(h^5), \quad (21)$$

is the sum of the error of the third order Runge-Kutta scheme and the error of the central differences \tilde{u}' and \tilde{u}'' , which corresponds to the error in the velocity extrapolation. It is important to remark, that even though the central differences preserved the order of the local truncation error of the Runge-Kutta scheme, the local truncation error in the extrapolation is proportional to the fourth power of the number of extrapolated time steps $(k+1)$. This means, the extrapolation can only be used at slow moving particles, where a small value of $u'''(0)$ balances the error.

The two-times frame algorithm is as follows: we compute the extrapolated new position χ_{k+1} . If the normed displacement of the particle from the position at the local initial particle time is less than the accuracy parameter e

$$\frac{|\chi_{k+1} - \chi_0|}{\delta} < e, \quad (22)$$

then χ_{k+1} is accepted as the new particle position. Otherwise we compute χ_{n+1} with full velocity evaluation, update \tilde{u}' and \tilde{u}'' , and reset the local particle time to zero. The normalization of the particle displacement with the particle core size δ recognizes this parameter as a measure of resolution. A more appropriate scheme would evaluate the third derivative of the velocity, in order to use the error estimate (21) to select the particles to be moved using the interpolation. The cost of this scheme could be higher since more velocity evaluations would be required for the evaluation of the third derivative of the velocity.

One example of how this scheme may be used to increase accuracy in selected regions with less expense is presented in Fig. 3. In this case we present the inter-filament distance s_d and the maximum curvature $(\kappa \delta)_{max}$ vs. time, as we decrease the time step from $dt = 10e-4$ to $5e-4$ for a collapse run with the accuracy parameter $e = 0.1$. In the case of the inter-filament distance s_d , convergence is observed for the two smaller time steps and a larger error for the coarse time step. The error is decreased in the half time step by multiplying the CPU time by a factor of 1.4 instead of 2 required by the case with no extrapolation.

4 Filament surgery

Surgery for vortex contours (2D) and filaments (3D) has been used in the past. In 2D, it was used by Berk and Roberts [5] in the "water-bag" model for Vlasov plasmas and also by Overman and Zabusky [19] in Contour Dynamics [26] for the 2D Euler equations with simple topologies. The Contour Dynamics/Contour Surgery algorithm by Dritschel [9], based on proximity and curvature, has been applied successfully in large scale simulations with complex topologies. For 3D flows, Leonard [14] performed filament surgery based on proximity and orientation (antiparallel configuration). More recently, Chorin [7, 8] introduced a hairpin removal and filament surgery algorithm based on proximity and orientation in the theoretical framework of the "infinite temperature" state. (See also Schwarz, Phys. Rev B 31 (1985) 5782).

In previous studies of vortex collapse [10, 11], it is found that the energy density of the collapsing region in the filament tends to zero as the collapse time is approached. This signature of collapse occurs despite the observed increase in the self-energy. The reason for this behavior is that the antiparallel region of the vortex filament produces also a peak-like contribution that cancels the self-energy growth. Collapsed regions in the filament don't contribute much to the total energy of the flow (they are low or zero energy density regions), but they produce a very large growth of vortex elements in algorithms with variable spatial resolution, due to the large vortex stretching present. The growth of the number of particles can be so large, that the simulation has to be stopped because of the computational expense. In order to regularize the particle growth it is necessary to introduce a filament surgery algorithm to eliminate the collapsed portions of the vortex filaments.

In our algorithm the energy

$$E = \frac{1}{2} \sum_{j=1}^{n_f} \Gamma_j \int_{C_j} \mathcal{E}(s_j) ds_j \quad (23)$$

is computed using the approximation of the energy density

$$\tilde{\mathcal{E}}(s) = \frac{d\chi}{ds} \cdot \Psi(s), \quad (24)$$

where Ψ is defined in Eq. 6. After computing the energy density in the filaments, we identify the regions below an specified energy density threshold d . The next step is to find pairs of reconnecting regions by checking orientation of the thresholded regions. This permits the identification of "local" and "nonlocal" collapse regions, which are dealt with differently for bookkeeping purposes. The last step is to eliminate low energy density regions by reconnecting and deleting particles (bookkeeping operations). The variable resolution subroutine implemented in the past provides some rounding of the edges left by the surgery as it interpolates new grid points with a cubic spline. At present we restrict reconnection to regions in the same filament. This is only to simplify the bookkeeping. In the future we will remove this restriction, leaving only the condition that the flow of vorticity through the vortex filaments remains continuous (vorticity must be divergence free). This will be done by reconnecting filaments only if they have the same circulation.

Surgery is a regularization that produces punctuated or intermittent dissipation. This means, it operates only when a collapse event has occurred, and at the level of the small scales generated by this event. We are still investigating how the rate of energy dissipation obtained by this algorithm scales with respect to the enstrophy of the flow. Another related aspect we are examining, corresponds to the energy density threshold to be used, which we believe should be the minimum energy density at the validity limit (determined by the local core size) of the simulation.

Our numerical results are as follows. In Fig. 4 we present the effect obtained by using different energy density thresholds d for a single-filament collapsing vortex ring at $t = 0.23$, with the constant

core model [10]. Whereas for $d = 0$ there is practically no small scales removal, these are removed more efficiently as the energy density threshold is increased. It can be noticed that the only region affected by the surgery corresponds to the collapse region. In Fig. 5(a), we observe how the growth of the number of particles N_p is reduced as the energy density threshold is increased. The largest particle growth corresponds to the no surgery case. In Fig. 5(b) we see that the inter-filament distance S_d increases as vortex reconnection takes place. This occurs earlier in time for higher energy density threshold d .

In Figs. 6(a) and (b) we have the non-zero component of the linear impulse and the energy for the simulation with energy density threshold $d = 0.03$. In parts (c) and (d) of the same figure we observe the inter-filament distance and the maximum normalized curvature $(\kappa \delta)_{max}$ vs. time respectively. The continuous line corresponds to the combination of the surgery, with four point smoothing of regions with high normalized curvature $\kappa \delta > 2$. Even though the appearance of the curves is smoother for this case, the cost in terms of the energy loss is higher. In part (d) of the figure we observe that as the reconnection takes place, the surgery algorithm leaves high curvature regions in the filaments. These regions are eliminated by the surgery only as they evolve into hairpins (with the pairing characteristic of antiparallel regions). In the other hand, the curvature smoothing eliminates the high curvature regions more effectively. We are examining currently the meaning of these results in terms of energy dissipation and expect to have quantitative results in the near future.

We started applying the filament surgery to multi-filament simulations with the constant core model. The no surgery case is presented in Fig. 7 for a ring with 24 filaments. The same case at the energy density threshold $d = 0.02$, with no curvature smoothing, is presented in Fig. 8. A more detailed view of the reconnection process at $t = 0.2$ and $t = 0.22$ is presented in Fig. 9. The diagnostics box surrounds the region with the maximum normalized vortex stretching magnitude. The last time shown corresponds to the end of the reconnection. The isosurfaces correspond to the vorticity magnitude. In Fig. 10 we have the non-zero component of the linear impulse and the energy vs. time for the multi-filament simulation. In Fig. 11(a) we observe how the surgery algorithm effectively regularizes the very steep growth of the number of particles observed in the no surgery case. In part (b) of the same figure we observe the evolution of the maximum curvature of the filaments for these simulations, which were performed without the curvature smoothing process.

5 Conclusions

In this work we first reduce the computational complexity of our simulations via a high performance parallel implementation of the Barnes-Hut method. Then we introduce a temporal extrapolation scheme that permits higher resolution in selected regions at lower computational expense. Finally, we introduce a filament surgery algorithm based on the energy density. This is an effective regularization that produces punctuated dissipation, as it eliminates small scales whenever they are generated during collapse events.

This work is part of the Hypercomputing and Design (HPCD) project, supported by ARPA, contract DABT-63-93-C-0064. Parallel simulations were performed on the CM5 at NCSA. Future implementations will be carried out on the SP2 at CTC.

References

- [1] Aarseth, S.J., “Dynamical Evolution of Clusters of Galaxies, I,” *Monthly Notices Roy. Astronomical Soc.*, **126** (3), pp. 223–255, 1963.
- [2] Ahmad, A. and Cohen, L., “A Numerical Integration Scheme for the N -Body Gravitational Problem,” *J. Comp. Phys.*, **12**, pp. 389–402, 1973.
- [3] Band, W., Introduction to Mathematical Physics, University Physics Series D, Van Nostrand Company, Inc., Princeton, New Jersey, pp. 191–197, 1959.
- [4] Barnes, J. and Hut, P., “A Hierarchical $O(N \log N)$ Force-Calculation Algorithm,” *Nature*, **324**, 1986.
- [5] Berk, H. and Roberts, K., “The Watter-Bag Model,” *Meth. Comp. Phys.*, **9**, pp. 87, 1970.
- [6] Bhatt, S., Liu, P., Fernandez, V. and Zabusky, N., Tree Codes for Vortex Dynamics: Application of a Programming Framework, Workshop on Solving Irregular Problems on Parallel Machines, International Parallel Processing Symposium, Santa Barbara, 1995.
- [7] Chorin, A.J., “Hairpin Removal in Vortex Interactions,” *J. Comp. Phys.*, **91**, pp. 1–21, 1990.
- [8] Chorin, A.J., “Hairpin Removal in Vortex Interactions II,” *J. Comp. Phys.*, **107** (1), pp. 1–9, 1993.
- [9] Dritschel, D.G., “Contour Surgery: A Topological Reconnection Scheme for Extended Integrations using Contour Dynamics,” *J. Comp. Phys.*, **77**, pp. 240–266, 1988.
- [10] Fernandez, V.M., Zabusky, N.J. and Gryanik, V.M., “Vortex Intensification and Collapse of the Lissajous-Elliptic Ring: Single and Multi-Filament Biot-Savart Simulations and Visiometrics,” *J. Fluid Mech.*, **299**, pp. 289–331, 1995.
- [11] Fernandez, V.M., Zabusky, N.J. and Gryanik, V.M., “Near-Singular Collapse and Local Intensification of a Lissajous-Elliptic Vortex Ring: Non-Monotonic Behavior and Zero-Approaching Local Energy Densities,” *Phys. Fluids A*, **6** (7), pp. 2242–2244, 1994.
- [12] Jernigan, J.G. and Porter, D.H., “A Tree Code with Logarithmic Reduction of the Force Terms, Hierarchical Regularization of all Variables, and Explicit Accuracy Controls,” *Astrophysical J. Suppl. Series*, **71**, pp. 871–893, 1989.
- [13] Knio, O.M. and Ghoniem, A.F., “Numerical Study of a Three-Dimensional Vortex Method,” *J. Comp. Phys.*, **86**, pp. 75–106, 1990.
- [14] Leonard, A., “Numerical Simulation of Interacting, Three-Dimensional Vortex Filaments,” *Proc. 4th International Conference in Numerical Methods in Fluid Dynamics*, number 35 in Lecture Notes in Physics, Springer-Verlag, pp. 245–250, 1975.
- [15] Leonard, A., “Computing Three-Dimensional Incompressible Flows with Vortex Elements,” *Ann. Rev. Fluid Mech.*, **17**, pp. 523–559, 1985.
- [16] Liu, P., “The Parallel Implementation of N -Body Algorithms,” DIMACS Technical Report, 94-27, 1994.

- [17] Majda, A.J., “Vorticity, Turbulence and Acoustics in Fluid Flow,” *SIAM Review*, **33 (3)**, pp. 349–388, 1991.
- [18] McMillan, S.L.W. and Aarseth, S.J., “An $O(N \log N)$ Integration Scheme for Collisional Stellar Systems,” *Astrophysical J.*, **414**, pp. 200–212, 1993.
- [19] Overman, E.A. and Zabusky, N.J., “Evolution and Merger of Isolated Vortex Structures,” *Phys. Fluids*, **25 (8)**, pp. 1297–1305, 1982.
- [20] Salmon, J.K., Parallel Hierarchical N-Body Methods, Ph.D. Thesis, California Institute of Technology, Pasadena, California 91125, 1990.
- [21] Salmon, J.K. and Warren, M.S., “Skeletons from the Treecode Closet,” *J. Comp. Phys.*, **111**, pp. 136–155, 1994.
- [22] Salmon, J.K., Warren, M.S. and Winckelmans, G.S., “Fast Parallel Tree Codes for Gravitational and Fluid Dynamical N -Body Problems,” *Int'l. Supercomputer Appl.*, 8.2, 1994.
- [23] Schwartz, L., *Analyse Mathématique*, Vol. 1, Chap. III, Hermann, Paris, pp. 248–265, 1967.
- [24] Sundaram, S., Fast Algorithms for N -Body Simulations, Ph.D. Thesis, Graduate School, Dept. of Computer Science, Cornell University, Ithaca, N.Y., August 1993.
- [25] Winckelmans, G.S., Salmon, J.K., Leonard, A. and Warren, M.S., “Particle and Panel Methods: Fast Tree-Code Solvers with Active Error Control for Arbitrary Distributions/Geometries,” *1st Annual Forum on Vortex Methods for Engineering Applications*, Sandia National Laboratories, Albuquerque, NM, pp. 23–43, 1995.
- [26] Zabusky, N.J., Hughes, M.H. and Roberts, K.V., “Contour Dynamics for the Euler Equations in Two Dimensions,” *J. Comp. Phys.*, **30**, pp. 96–106, 1979.

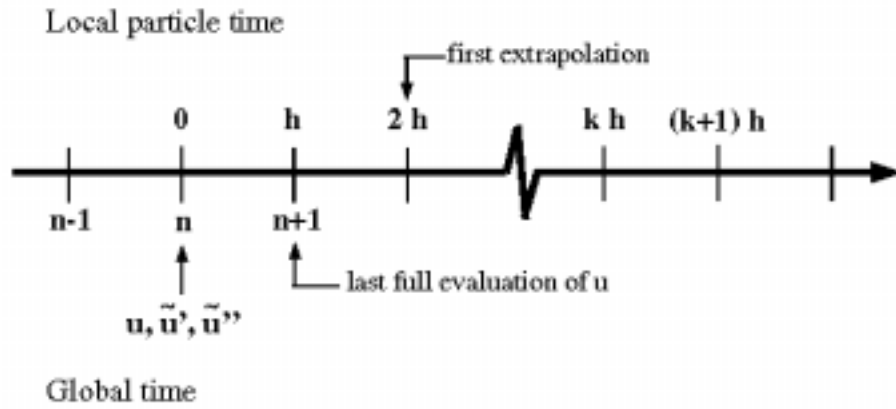


Figure 1: Global and local particle time frames.

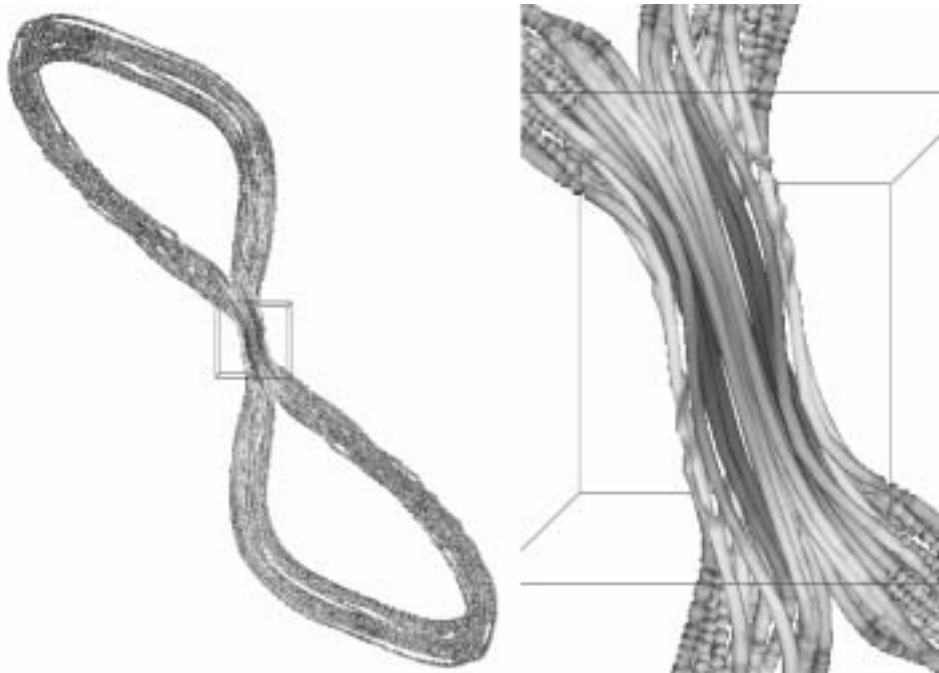


Figure 2: High resolution ($N_p = 319\,488$ particles) simulation of vortex collapse.

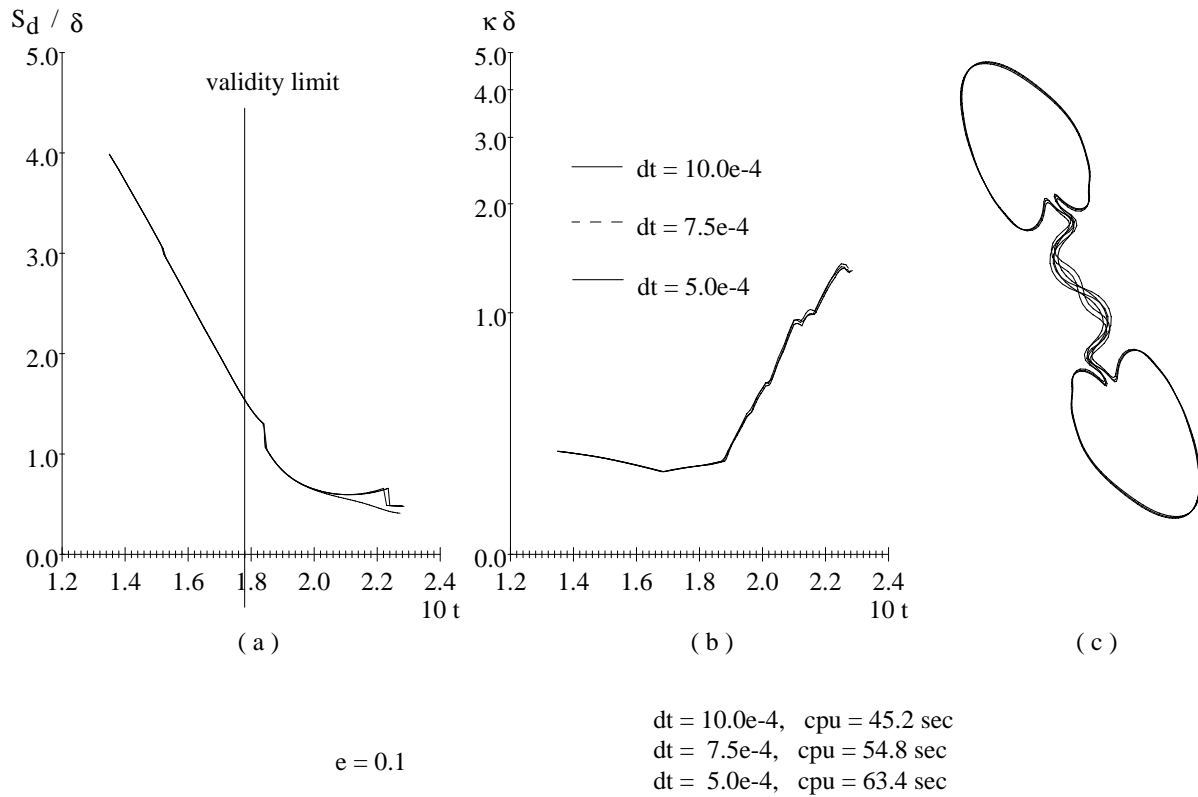


Figure 3: (a) Inter-filament distance S_d vs. time, (b) maximum curvature $(\kappa \delta)_{max}$ vs. time and (c) top view of vortex ring for different time steps. Use of time extrapolation with $e = 0.1$ produces savings in cpu time as the time step is decreased.

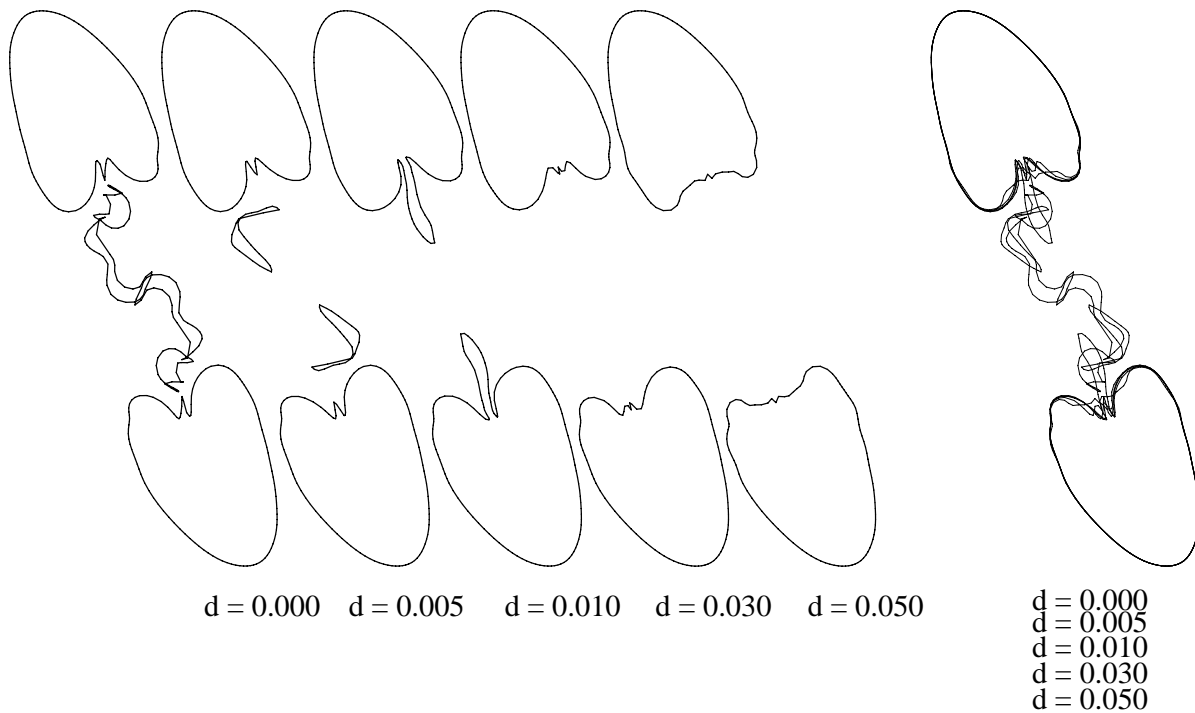


Figure 4: Surgery of single-filament ring at different energy density thresholds d .

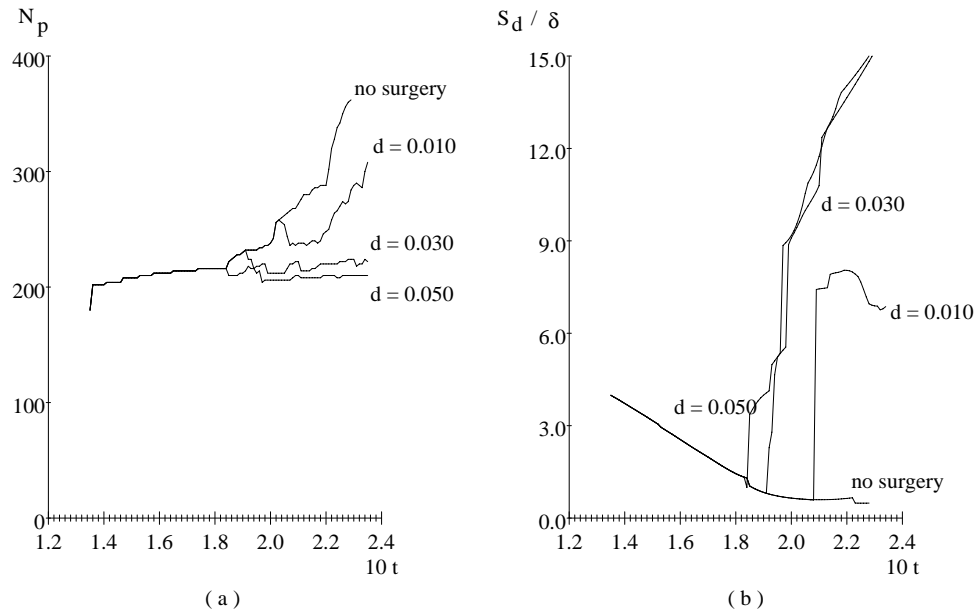


Figure 5: (a) Evolution of the growth of the number of particles N_p vs. time for the single-filament ring with surgery for different energy density thresholds d . (b) Inter-filament distance S_d vs. time

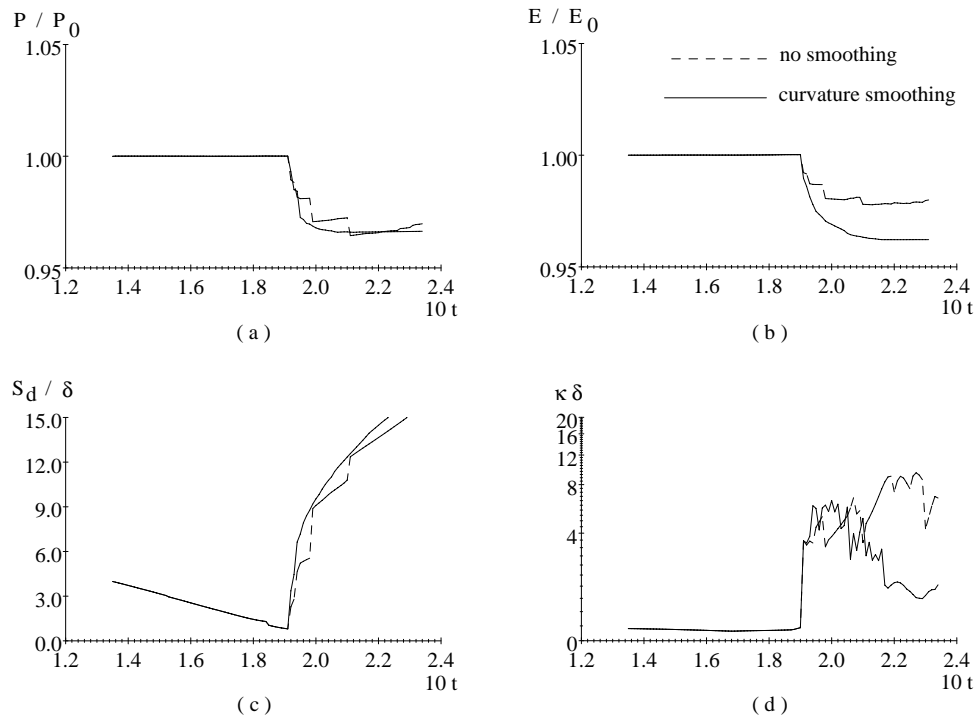


Figure 6: (a) Non-zero component of linear impulse P and (b) energy E are normalized with the initial values P_0 and E_0 respectively. (c) Inter-filament distance S_d vs. time. (d) maximum normalized curvature $(\kappa \delta)_{max}$ vs. time. The continuous line combines surgery with smoothing of high curvature regions.

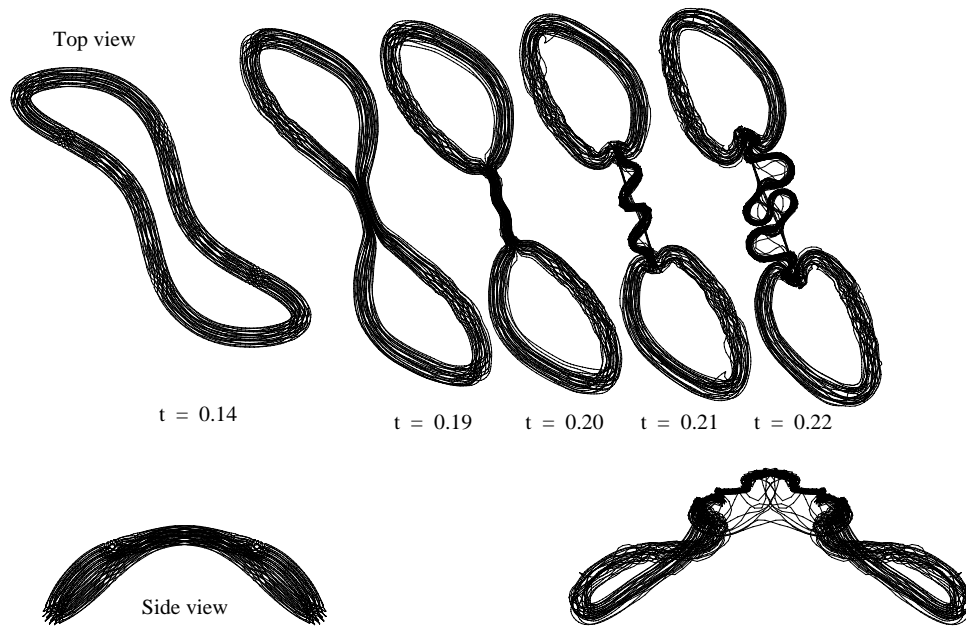


Figure 7: Evolution of multi-filament ring with no surgery.

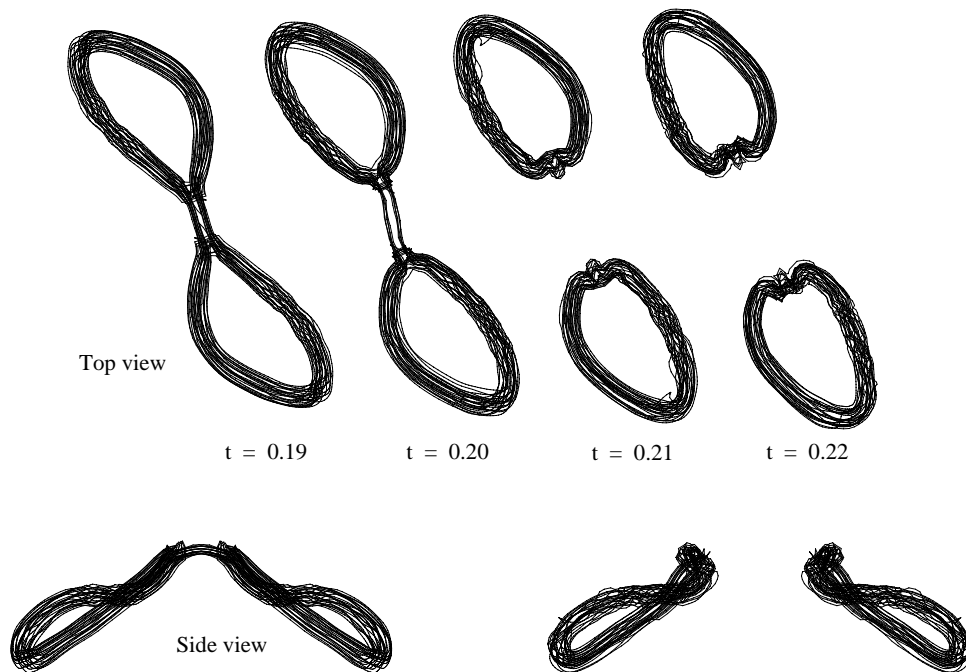


Figure 8: Multi-filament ring with surgery at the energy density threshold $d = 0.02$ (no curvature smoothing).

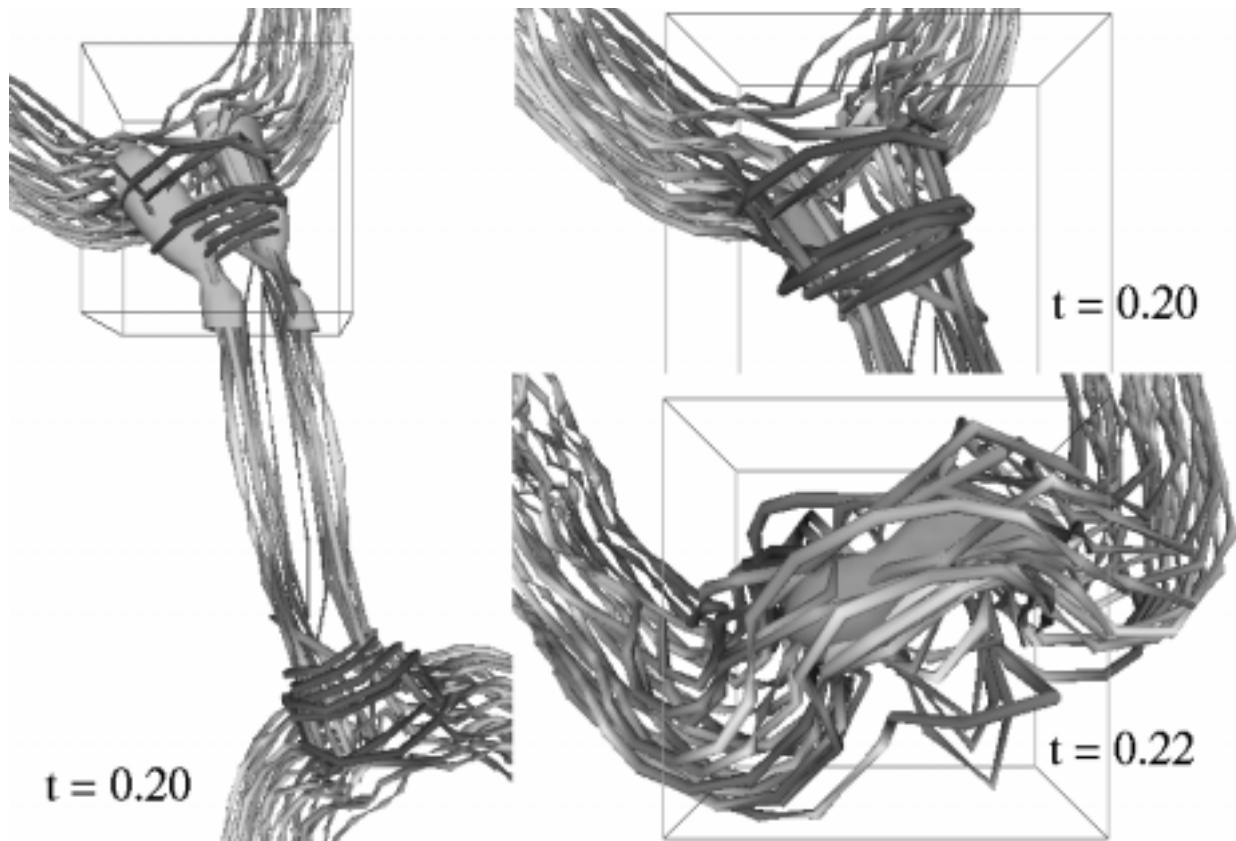


Figure 9: Diagnostics box in region with maximum normalized vortex stretching magnitude at two different times in the reconnection process.

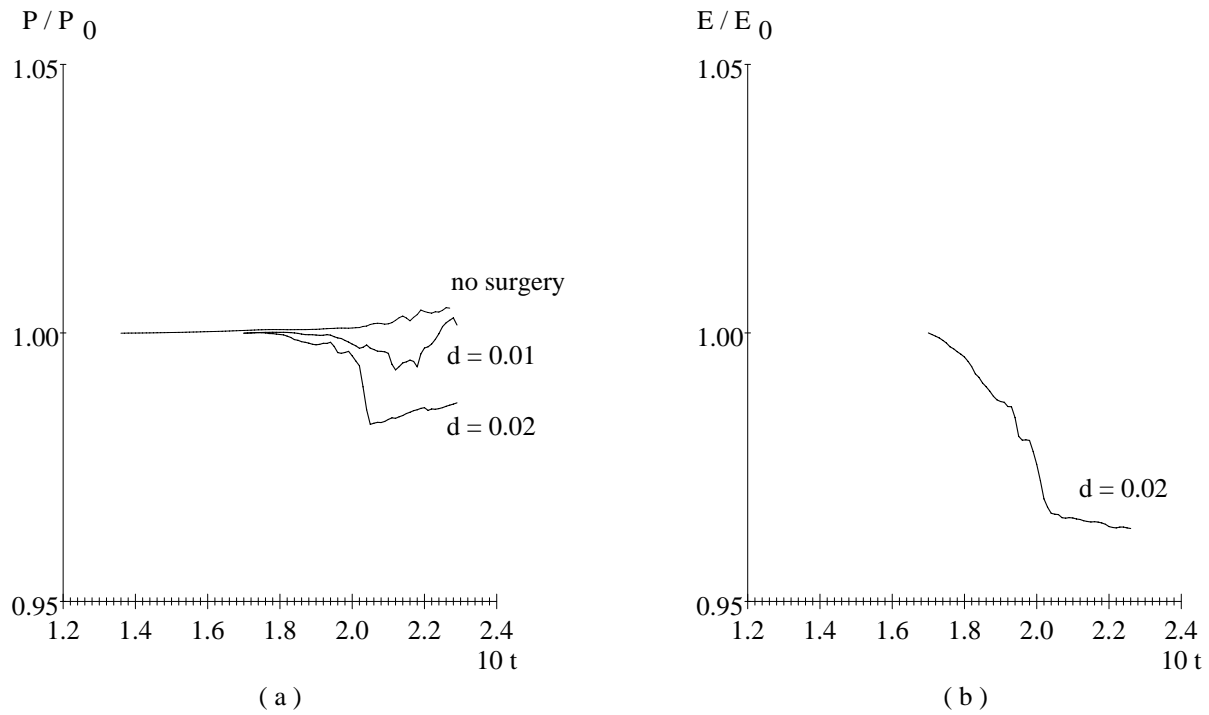


Figure 10: (a) Linear impulse (non-zero component) and (b) energy for the vortex reconnection multi-filament simulation.

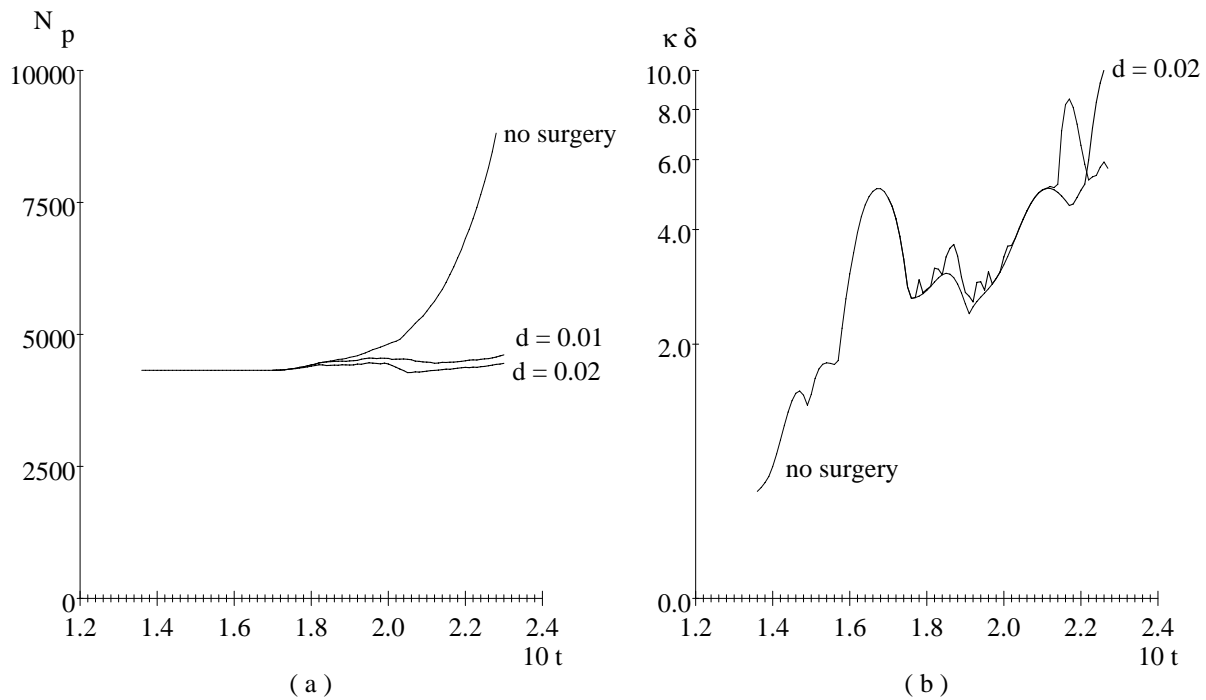


Figure 11: (a) Growth of the number of particles N_p vs. time for the multi-filament ring with and without surgery. (b) Maximum normalized curvature $(\kappa \delta)_{max}$ vs. time.