

# Supporting Information for “Theory of flow-induced covalent polymer mechanochemistry in dilute solutions”

Etienne Rognin, Niamh Willis-Fox, and Ronan Daly

## 1 Supporting dataset and software

The simulation dataset and post-processing code used for this study are available at <https://doi.org/10.17863/CAM.101595> under open access license CC BY 4.0.

**Main file** The *Jupyter Notebook* `Supporting information.ipynb` can be used to run the data analysis and recreate the figures of the paper. Alternatively, a `pdf` file provides a view of the notebook outputs without having to install and run the scripts.

**Raw data** The folder `bead-rod_dataset` contains the results of bead-rod model simulations. For each simulation there is binary `.npz` Python file containing the data, and a `.json` text file containing metadata (such as date of the simulation, parameters...). The data is imported using `numpy.load` function which creates a Python dictionary for each simulation file. This dictionary contains the following labels:

<code>t</code>	the time axis.
<code>gradU</code>	the time series of velocity gradients used as forcing terms in the bead-rod simulation.
<code>g_max</code>	the time series of the maximum tensile force, for each molecule of the simulation ensemble.
<code>i_max</code>	the time series of the positions of the maximum force in the chain (not used in this study)
<code>g_12</code>	the time series of the tensile force at the center of the chain, for each molecule.
<code>A_average</code>	the time series of the average conformation tensor (second-order moment of the end-to-end vector). Used in section 4 for model validation.

Values are dimensionless (normalisation coefficients are described in the next section).

## 2 Polymer models

Backbone tensions are simulated using coarse-grained molecular models of polymers. In this study we use the freely-jointed bead-rod model: a polymer molecule is represented by a series of beads and links. Friction with solvent molecules is located at beads only and links

are rigid rods. There can also be conservative interactions between non-contiguous beads, for example excluded volume effects, and dissipative hydrodynamic interactions (HI).

Assuming the chain has  $N$  links,  $N + 1$  beads, then for each bead  $i$ , the position vector,  $\mathbf{x}_i$ , relative to the centre of friction is evolved by a time step  $dt$  using the following stochastic differential equation:

$$d\mathbf{x}_i = \left( \mathbf{x}_i \nabla \mathbf{u} + \sum_{j=1}^{N+1} M_{ij} \mathbf{f}_j \right) dt + \sqrt{2k_B T} \sum_{j=1}^{N+1} B_{ij} d\mathbf{w}_j \quad (1)$$

where  $\nabla \mathbf{u}$  is the gradient of the solvent velocity field imposed far from the molecule (background flow),  $M_{ij}$  is the pair-wise mobility tensor between beads  $i$  and  $j$ ,  $\mathbf{f}_j$  is the net force on bead  $j$  including links tension and conservative interactions if present,  $k_B$  is the Boltzmann constant,  $T$  is the temperature,  $B_{ij}$  is a matrix so that  $M_{ij} = \sum_{k=1}^{N+1} B_{ik} B_{jk}^T$ , and  $d\mathbf{w}_j$  is a Wiener process. In this notation, the block-matrix  $M$  also includes self-mobility (or drag), which is given by:

$$M_{ii} = \frac{1}{\zeta_i} \quad (2)$$

where  $\zeta_i$  is the bead friction coefficient. The rest of  $M$  is null if HI are ignored (*free-draining* approximation), or otherwise given by the Rotne-Prager-Yamakawa hydrodynamic tensor:

$$M_{ij} = \begin{cases} \frac{1}{8\pi\eta|\mathbf{r}_{ij}|} \left( \left( 1 + \frac{2a^2}{3|\mathbf{r}_{ij}|^2} \right) \mathbf{I} + \left( 1 - \frac{2a^2}{|\mathbf{r}_{ij}|^2} \right) \frac{\mathbf{r}_{ij}\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^2} \right) & \text{if } |\mathbf{r}_{ij}| > 2a \\ \frac{1}{6\pi\eta a} \left( \left( 1 - \frac{9|\mathbf{r}_{ij}|}{32a} \right) \mathbf{I} + \frac{3}{32a|\mathbf{r}_{ij}|} \mathbf{r}_{ij}\mathbf{r}_{ij} \right) & \text{if } |\mathbf{r}_{ij}| \leq 2a \end{cases} \quad (3)$$

where  $\mathbf{r}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ ,  $a$  is the beads hydrodynamic radius and  $\eta$  is the solvent viscosity.

For rigid links, rod tension is solved by a fixed-point method derived from the rods length constraint, as detailed in our previous work.<sup>1</sup> If we note  $b$  the rod length (same length for all rods), a dimensionless model is derived by normalizing lengths with  $b$ , forces with  $k_B T/b$  and time with  $\zeta b^2/k_B T$ .

When there are no HI nor non-contiguous beads interactions, the mobility block-matrix  $M$  (in equation 1) is sparse and evolving a molecule is rather computationally cheap as it requires solving tridiagonal matrices. When HI are included, the strength of hydrodynamic interaction is set by the value of  $a$ , larger beads yielding stronger interaction. It is more common to use  $h = \pi^{-\frac{1}{2}} a$  and its dimensionless counterpart,  $h^*$ . In the case of infinite Weissenberg number, Brownian forces are negligible and therefore there is no need to compute the matrix  $B$  from a factorization of  $M$ . However,  $M$  is dense and solving rod tensions becomes computationally expensive. Therefore only a reduced ensemble of molecules can be simulated.

## 2.1 Simulation software

We used our own bead-rod model solver which can be accessed at: <https://github.com/etiennerognin/DiluteBrownianDynamics>

## 3 Model validation

### 3.1 Flows definition

Here we describe how the model is validated in different scenarios. Note that all flows are obtained assuming Newtonian stress (ultra-dilute regime). Below are details on how the La-

<sup>1</sup>E. Rognin, N. Willis-Fox, T. A. Aljohani and R. Daly, *Journal of Fluid Mechanics*, 2018, **848**, 722–742. ([link](#))

grangian trajectories were obtained:

**4:1 contraction** This is a steady 2D-axisymmetric simulation done with OpenFOAM using a high-resolution mesh, at a Reynolds number of 100. Lagrangian trajectories are extracted at the post-processing step using Paraview. Initial seeds span from the centreline up to a small distance from the wall.

**Sonication** We follow the approach taken by Turetta and Lattuada,<sup>2</sup> and solve the Rayleigh-Plesset equation:

$$\rho \left( R\ddot{R} + \frac{3}{2}(\dot{R})^2 \right) = \left( p_0 + \frac{2\sigma}{R_0} - p_v \right) \left( \frac{R_0}{R} \right)^{3\kappa} + p_v - \frac{2\sigma}{R} - \frac{4\eta\dot{R}}{R} - p_0 - p_A(t) \quad (4)$$

where  $\rho$  is the solvent density,  $R$  is the bubble radius,  $p_0$ ,  $p_v$  and  $p_A(t)$  are the initial, saturation and gauge external pressure respectively,  $\sigma$  is the surface tension,  $R_0$  is the initial radius,  $\kappa$  is the polytropic coefficient of the gas, and  $\eta$  is the solvent viscosity. The equation can be rearranged as  $\dot{y} = f(y)$  with  $y = (R, V)$  and

$$\begin{aligned} \dot{R} &= V \\ \dot{V} &= -\frac{3}{2} \frac{V^2}{R} + \frac{1}{\rho R} \left( \left( p_0 + \frac{2\sigma}{R_0} - p_v \right) \left( \frac{R_0}{R} \right)^{3\kappa} + p_v - \frac{2\sigma}{R} - \frac{4\eta V}{R} - p_0 - p_A(t) \right) \end{aligned} \quad (5)$$

Bellow is a code snippet to solve this equation in Python using the `scipy solve_ivp` function, with physical parameters used in this study:

```
from scipy.integrate import solve_ivp

R0 = 10e-6      # m
pv = 3.9e3     # Pa
p0 = 1.01325e5 # Pa
eta = 0.56e-3  # Pa.s
sigma = 0.028  # N/m
rho = 940      # kg/m^3
kappa = 1.4
p1 = p0 + 2*sigma/R0 - pv
pA = 5e5       # Pa
f = 20e3       # Hz

def forcing(t):
    """External gauge pressure forcing term."""
    return pA*np.sin(2*np.pi*f*t)

def RayleighPlesset(t, y):
    """Rayleigh-Plesset equation where y=[R, R_dot]"""
    out = np.empty_like(y)
    out[0] = y[1]
    out[1] = -1.5*y[1]**2/y[0] + 1./(rho*y[0])*
        (p1*(R0/y[0])**3*(kappa) + pv - p0 - 2*sigma/y[0] - 4*eta*y[1]/y[0] - forcing(t)
        )
    return out

sol = solve_ivp(RayleighPlesset, (0, 100e-6), [R0, 0], max_step=1e-8)
```

<sup>2</sup>L. Turetta and M. Lattuada, *Industrial & Engineering Chemistry Research*, 2021, **60**, 10539–10550 (<https://doi.org/10.1021/acs.iecr.1c00233>)

To get Lagrangian strain rates, we define initial positions as:  $R_L(0) = \alpha R_0$  where  $\alpha = 1$  if the molecule is sitting at the bubble interface,  $\alpha > 1$  otherwise. Using conservation of the volume of the shell between  $R(t)$  and  $R_L(t)$ :

$$R_L^3(t) = R(t)^3 + (\alpha^3 - 1)R_0^3 \quad (6)$$

The velocity at the bubble interface is just  $\dot{R}(t)$ . Again, with conservation of volume, we have:

$$U(r)r^2 = \dot{R}(t)R^2 \quad (7)$$

and therefore:

$$\frac{\partial U}{\partial r} = -2 \frac{\dot{R}(t)R^2}{r^3} \quad (8)$$

Finally, the Lagrangian strain rate is:

$$\dot{\epsilon}(\alpha, t) = -2 \frac{\dot{R}(t)R(t)^2}{R(t)^3 + (\alpha^3 - 1)R_0^3} \quad (9)$$

We let  $\alpha$  vary from 1 to 10 to obtain different trajectories.

**Turbulence** Lagrangian trajectories are extracted from the open-access *Johns Hopkins Turbulence Databases*<sup>3</sup> using the *Channel flow* dataset. See our supporting dataset and software for further information.

**Inkjet** This flow is a 2D-axisymmetric simulation done with OpenFOAM using a high-resolution mesh and the *Volume-of-Fluid* method. The time-dependent flow rate is imposed at the inlet (far upstream of the nozzle) based on experimental measurements of a single-nozzle *Micro-fab* system. Lagrangian trajectories are extracted at a post-processing step using flow tracers in Paraview. See our supporting dataset for a video displaying the tracers as they pass the nozzle. The largest values of the Lagrangian velocity gradient are obtained by the tracer flowing near the wall, and by the tracer on the centreline which is caught in the filament breakup.

### 3.2 Normalization and rescaling step

For each flow scenario described above, the maximum dimensioned strain rate,  $\dot{\epsilon}_{\max}$ , is extracted as the maximum eigenvalue of the velocity gradient  $\nabla \mathbf{u}$  of the combined trajectories. For the bead-rod model simulations, the number of links,  $N$ , (or average number of links for a polydispersed ensemble), and the maximum desired Weissenberg number,  $Wi_{\max}$ , are used to build a normalized velocity gradient time series,  $\nabla \mathbf{u}^+$ , as follows:<sup>4</sup>

$$\nabla \mathbf{u}^+(t^+) = \frac{Wi_{\max}}{0.0142N^2 \dot{\epsilon}_{\max}} \nabla \mathbf{u}(t) \quad (10)$$

with respect to the normalised time,  $t^+$ :

$$t^+ = \frac{0.0142N^2 \dot{\epsilon}_{\max}}{Wi_{\max}} t \quad (11)$$

In this study, we have  $Wi_{\max} = 10^3$ .

<sup>3</sup><http://turbulence.pha.jhu.edu/datasets.aspx>

<sup>4</sup>E. Rognin, N. Willis-Fox, T. A. Aljohani and R. Daly, *Journal of Fluid Mechanics*, 2018, **848**, 722–742. ([link](#))

### 3.3 Results

Results for the four scenarios are shown below. See figure 5 in the main text for an explanation of one of the turbulence trajectory. Key:

$c_i$	Concentration of intact polymer chains (solid line: simulated from bead-rod model, dashed line: computed by the closure model).
$c_m$	Concentration of activated mechanophores (solid line: simulated from bead-rod model, dashed line: computed by the closure model).
non-spec	Concentration of non-specific scission (such that $c_i + c_m + \text{non-spec} = 1$ )
$\varepsilon_{\text{eff}}$	Effective positive strain (see main text, equation 12).
$\varphi$	Buckling strain (see main text, equation 11).
$w$	Instantaneous stain rate normalised by the critical strain rate for mechanophore activation.
$w_0$	Normalised threshold for the onset of non-specific scission (see main text, equation 8).

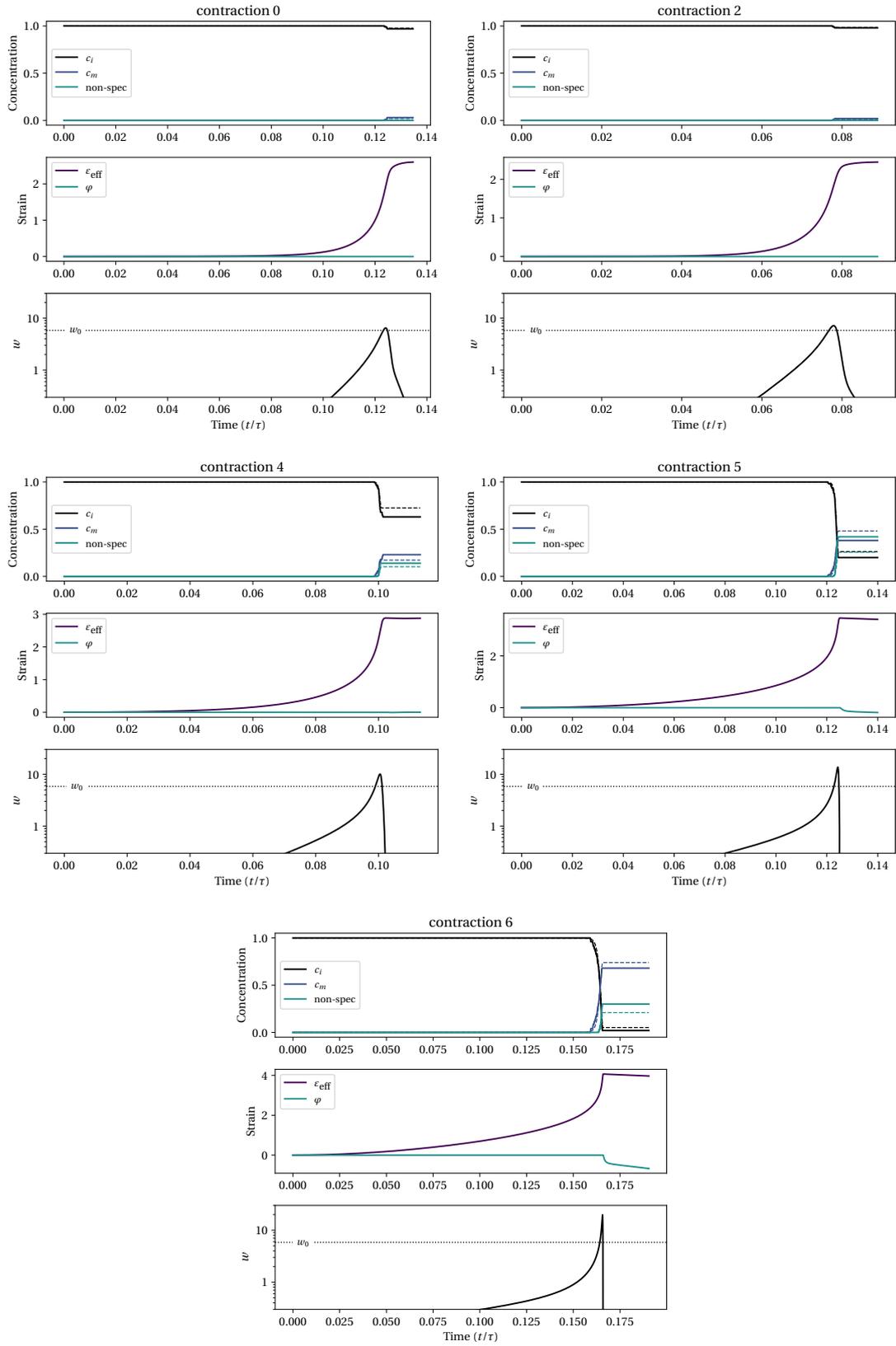


Figure 1: Results for the 4:1 contraction.

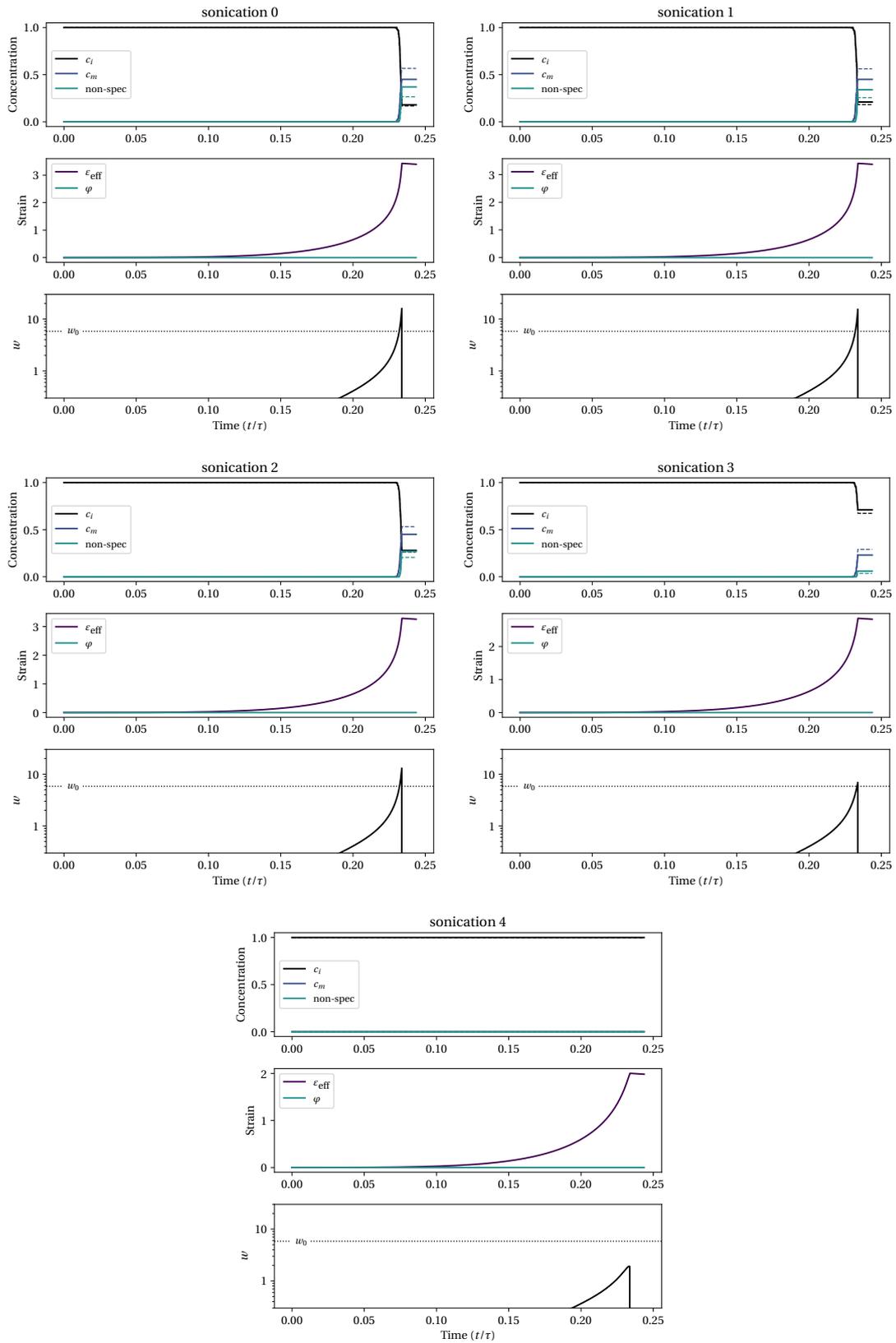


Figure 2: Results for sonication.

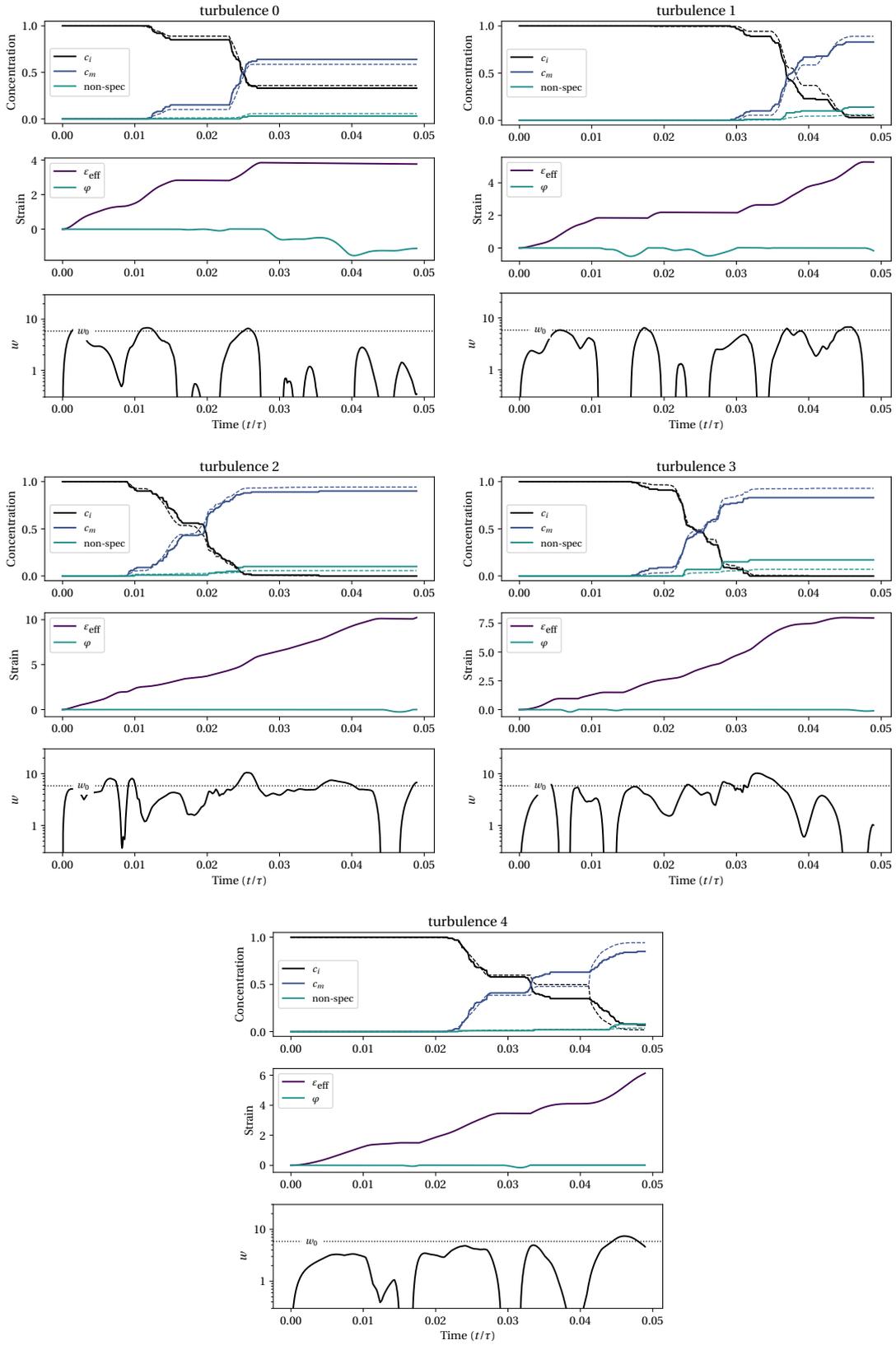


Figure 3: Results for turbulence.

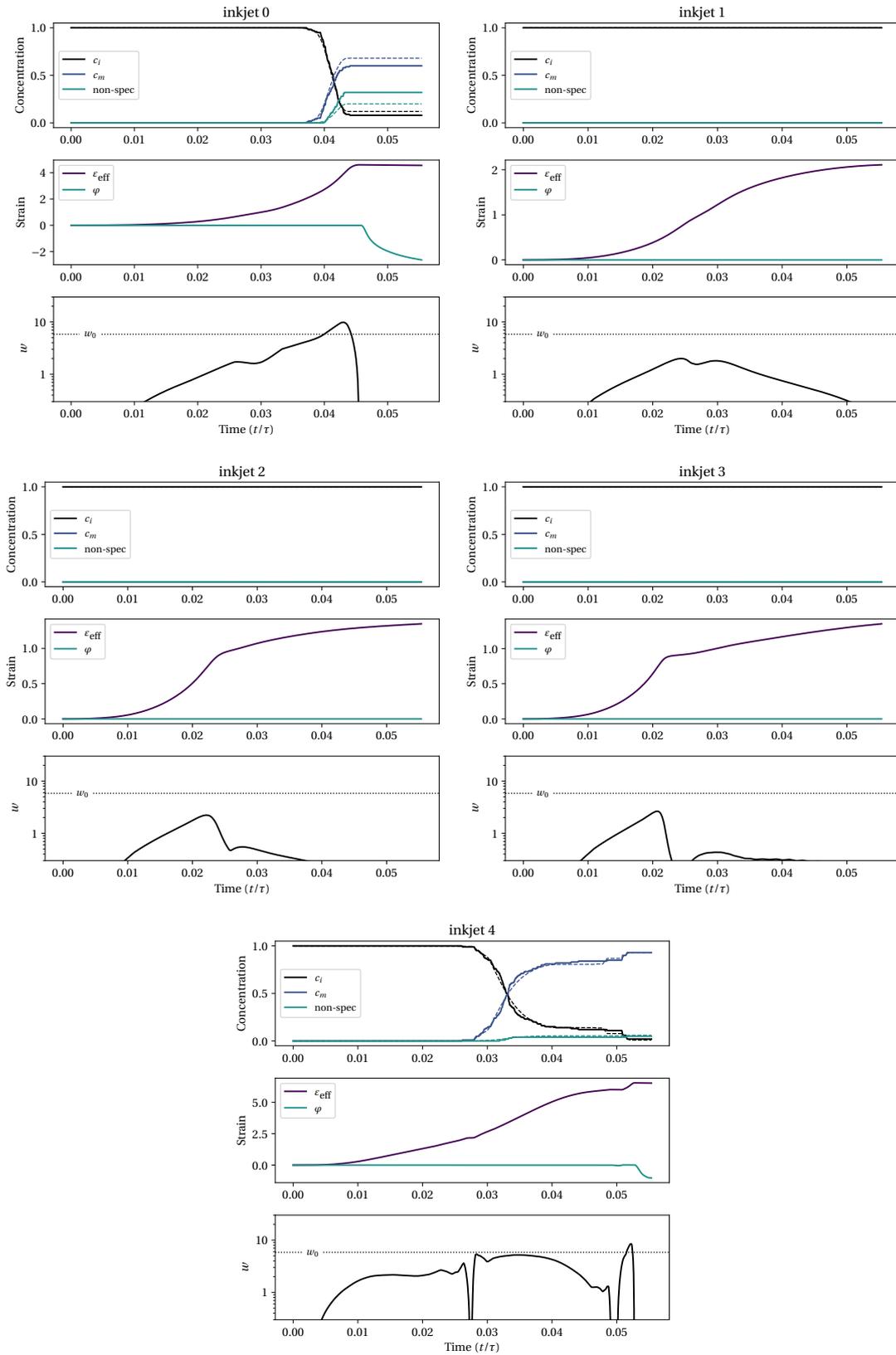


Figure 4: Results for inkjet.