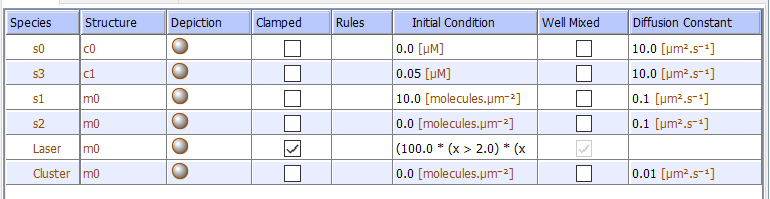
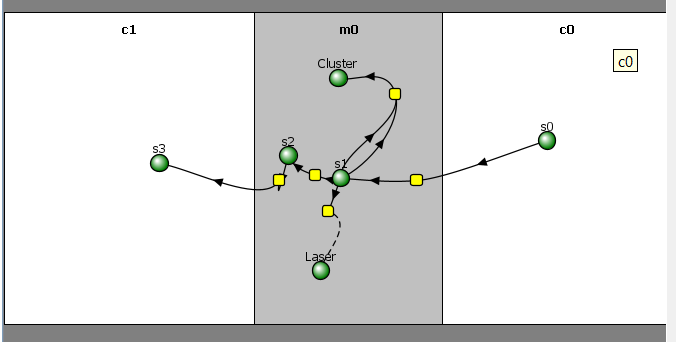
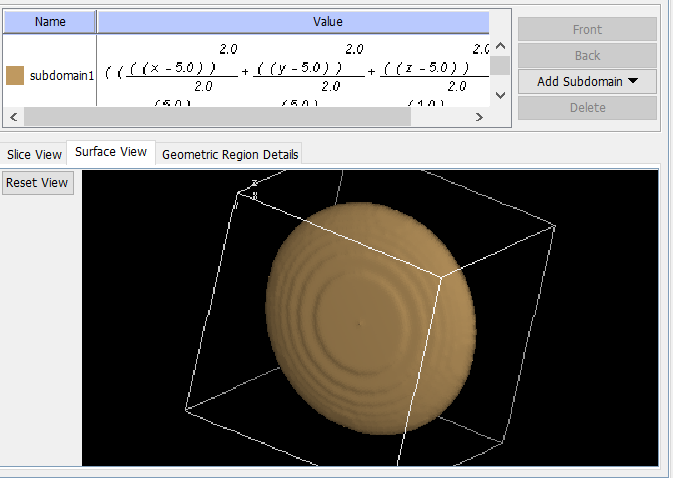
**Биофизика. Задание №8**

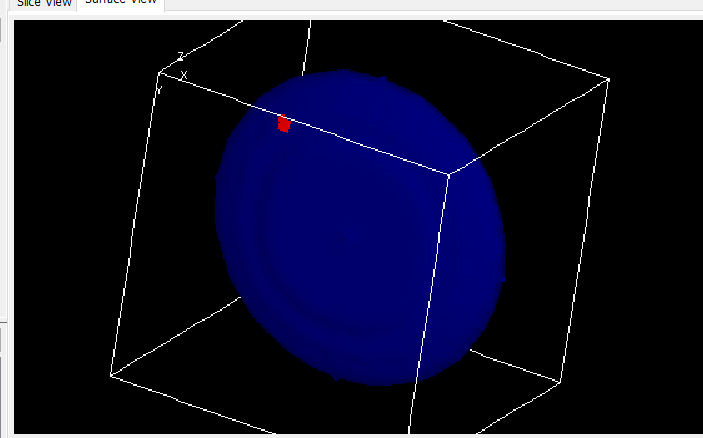
Создадим базовую модель FRAP в среде VCell.



Запускаем объёмную симуляцию действия лазера.

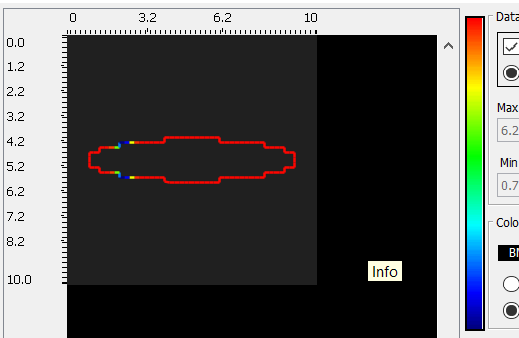


*В режиме slice view выбираем сечение и смотрим, как s1 уменьшается на мембране. Из-за объёмной симуляции лазер действует с обеих сторон.*

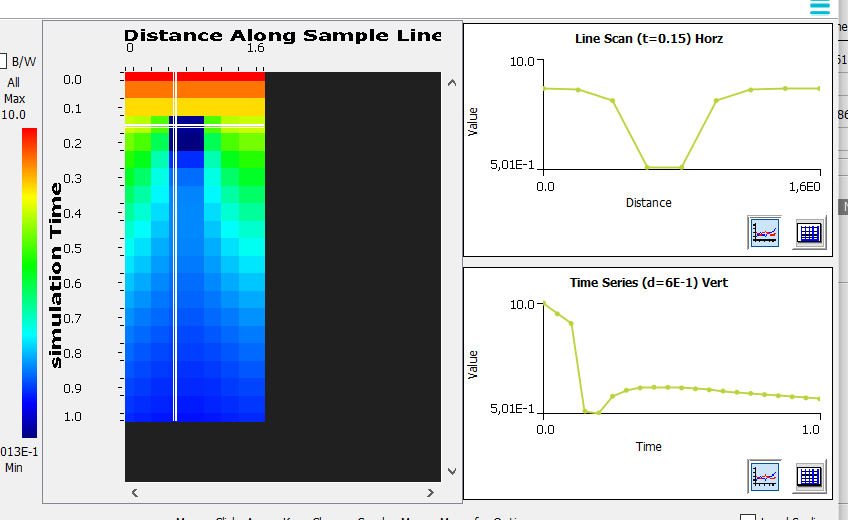


На верхнем графике виден импульс лазера, а на нижнем — типичная FRAP-кривая: после выгорания наблюдается восстановление сигнала за счёт диффузии.

**Можно включить режим подсчёта общего числа молекул. Сравним эффекты различных сочетаний кластеризации и лазера на количество s1:**



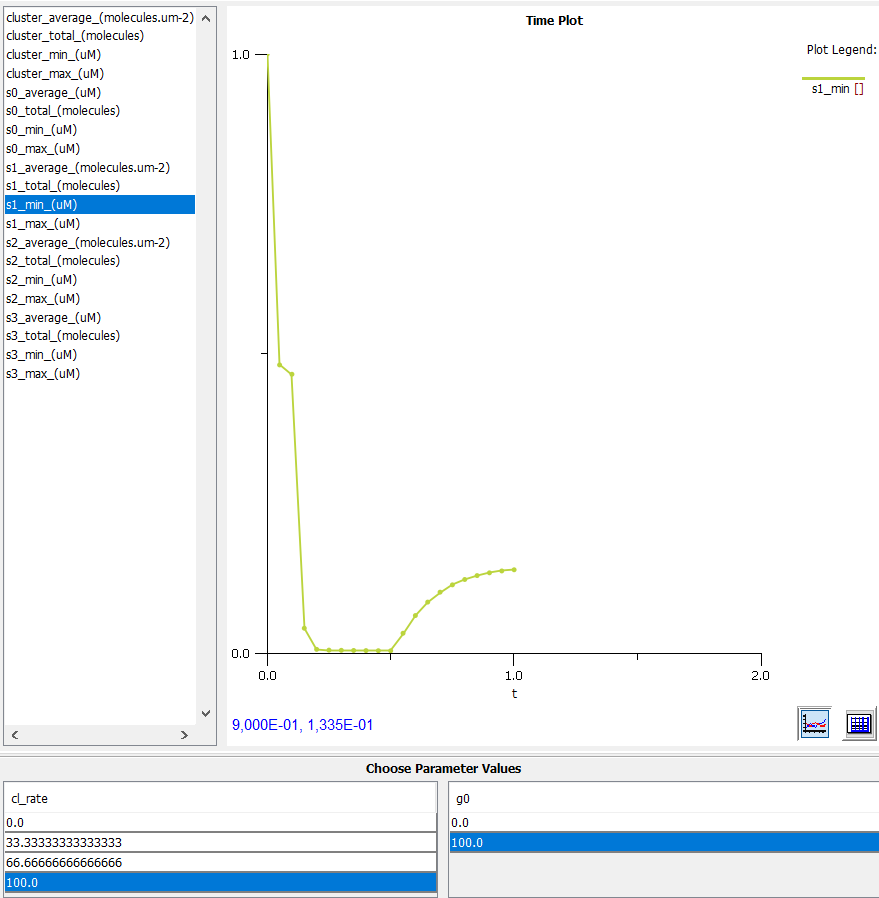
*Рассчитаем коэффициенты диффузии, в зависимости от уровня кластеризации.*



**Используем четыре уровня кластеризации и подберём значения D с помощью экспоненциальной аппроксимации:**

Подставим начальные приближения и найдём оптимальные параметры с помощью curve\_fit.

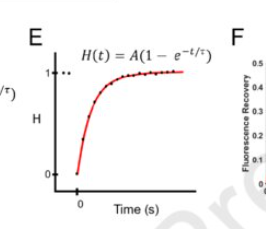
1. *Строим график: экспериментальные точки + модельная кривая.*
2. **Значения коэффициента диффузии D при разных степенях кластеризации:**
3. Кластеризация 0 → D = 1.83 мкм²/с



1. Кластеризация 66 → D = 1.38 мкм²/с

***Кластеризация 100 → D = 1.36 мкм²/с***

Вывод: увеличение кластеризации снижает диффузию, так как более крупные комплексы движутся медленнее. При высоких значениях кластеризации D стабилизируется.

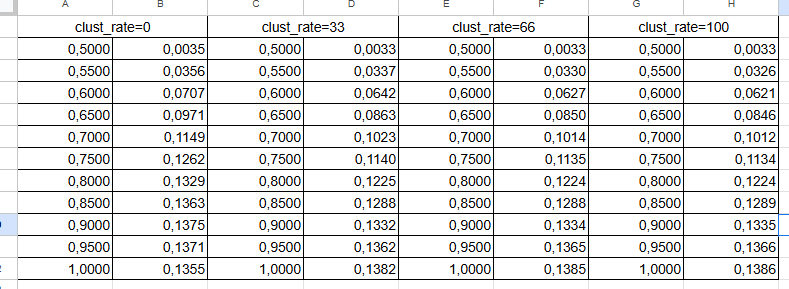


Здесь 1/tau=D/R^2

R=0.5 мкм - область действия лазера

Возьмем четыре точки для значений кластеризации. Будем аппроксимировать экспоненциальной зависимостью и получим D

Аппроксимацию будем делать при помощи кода. Здесь F\_data и t\_data - данные полученные из модели



def model(t, a, b, c):

return a \* (1 - np.exp(-b \* (t - c)))

initial\_guess = [0.14, 1.0, 0.0]

params, covariance = curve\_fit(model, t\_data, F\_data, p0=initial\_guess, maxfev=10000)

a\_opt, b\_opt, c\_opt = params

print(f"Оптимальные параметры: a = {a\_opt:.6f}, b = {b\_opt:.6f}, c = {c\_opt:.6f}")

# График

plt.scatter(t\_data, F\_data, label='Данные')

t\_fit = np.linspace(t\_data.min(), t\_data.max(), 100)

F\_fit = model(t\_fit, a\_opt, b\_opt, c\_opt)

plt.plot(t\_fit, F\_fit, 'r-', label=f'F(t) = {a\_opt:.4f}·(1 - exp(-{b\_opt:.4f}·(t - {c\_opt:.4f})))')

plt.xlabel('t')

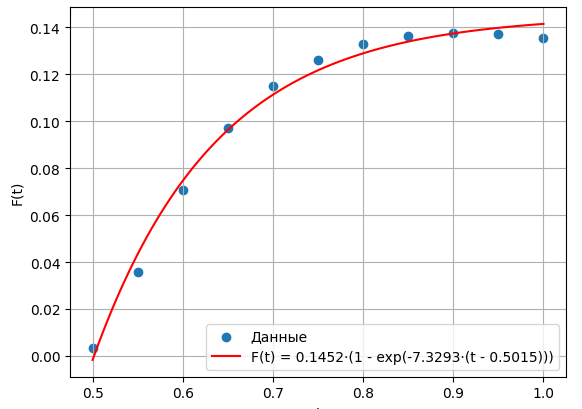
plt.ylabel('F(t)')

plt.legend()

plt.grid()

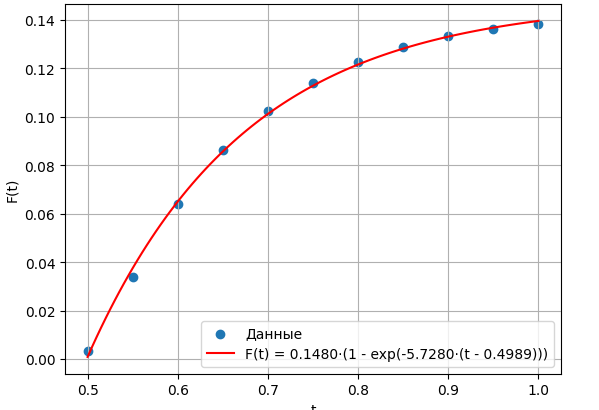
plt.show()

1. clust\_rate=0



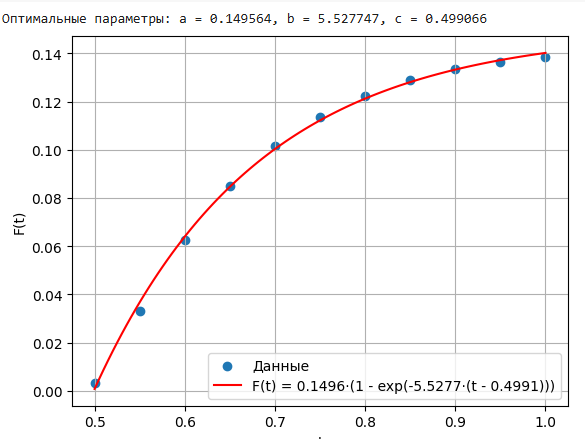
D = 7.3293\*0.5^2 = 1.83 мкм^2/c

1. clust\_rate=33



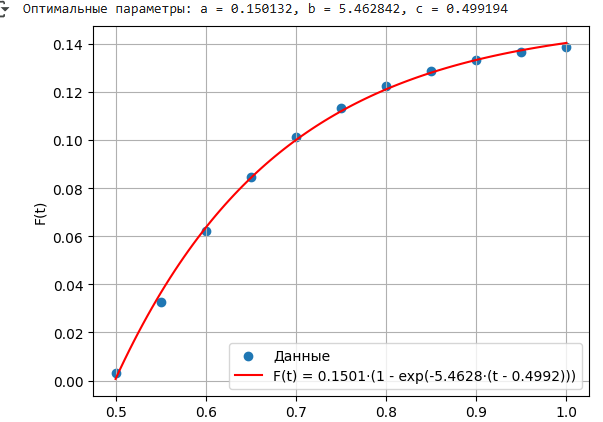
D = 5.728\*0.5^2 = 1.42 мкм^2/c

1. clust\_rate=66



D = 5.5277\*0.5^2 = 1.38 мкм^2/c

1. clust\_rate=100



D = 5.4628\*0.5^2 = 1.36 мкм^2/c

Чем больше значение кластеризации, тем меньше коэффициент диффузии. Это связано с тем, что крупные белковые комплексы медленнее диффундируют. С определенного значения кластеризации коэффициент диффузии практически не меняется.