

# SALMON VR - macOS User Manual

March 1, 2025

This document describes the installation and operation of SALMON VR - macOS version (hereinafter referred to as "this application").

## System Requirements

The system requirements for this application are as follows:

- macOS 14.5 or later
- Chip: Apple Silicon

## Installation

Extract the downloaded zip file to a location of your choice.

## Launch

Navigate to the extracted directory in Finder and double-click the `SALMON VR.app` file to launch the application.

Note: Due to macOS security features, apps downloaded from the web may not launch directly. Refer to the following instructions for troubleshooting.

## Troubleshooting

### Issue 1: "The developer cannot be verified"

Upon initial launch, you may encounter a warning dialog stating that the developer cannot be verified. If so, follow these steps:

1. Click OK to close the dialog.
2. Go to Apple menu > System Settings. In the sidebar, select "Privacy & Security". (You may need to scroll down.)
3. Go to "Security" and click "Open Anyway".
4. Enter your login password and click "OK".

### Issue 2: "The application cannot be opened"

If you see the message "The application "SALMON VR" cannot be opened", you may need to

add execution permission to the application. Follow these steps:

1. Click OK to close the dialog.
2. Open Terminal and navigate to the directory containing the application.
3. Execute the following command:

```
chmod a+x SALMON¥ VR.app/Contents/MacOS/SALMON¥ VR
```

### **Issue 3: "The file is damaged"**

If you see a message indicating the file is damaged, follow these steps:

1. Click Cancel to close the dialog.
2. Open Terminal and navigate to the directory containing the application.
3. Execute the following command:

```
xattr -d com.apple.quarantine "SALMON VR.app"
```

### **Issue 4: "Access to file" warning**

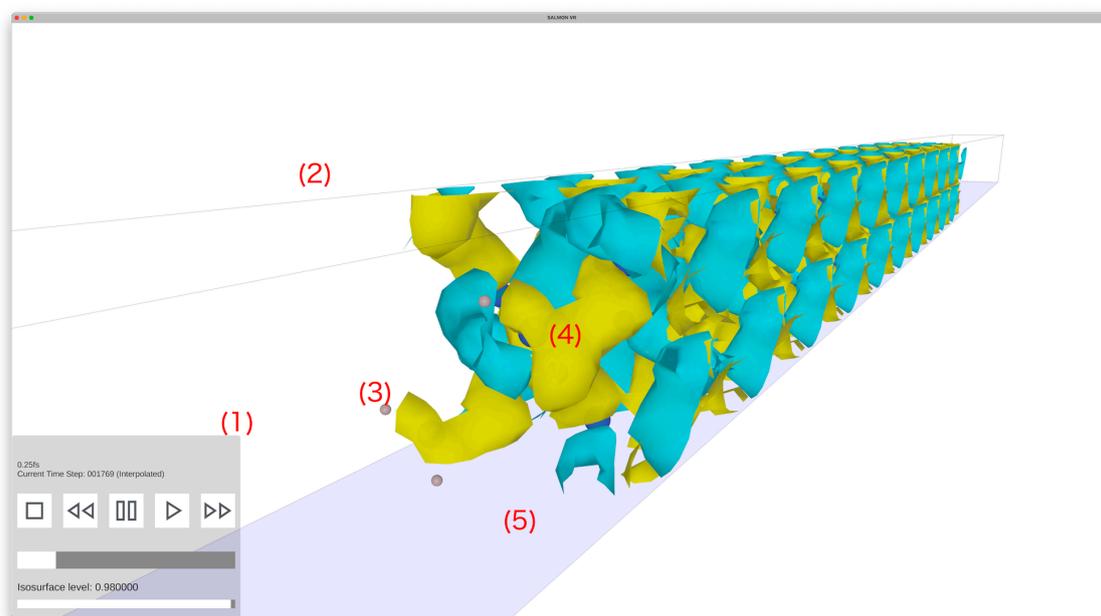
If you see a warning about the application accessing files in your Downloads folder, select "Allow".

## **Shutdown**

Close the application window by clicking the red button in the upper left corner.

## Interface Overview

The following is an explanation of the basic screen layout.



- (1) GUI Panel: Controls playback and isosurface level adjustment.
- (2) Calculation Space: The area inside the black lines.
- (3) Atoms: Atoms (e.g., hydrogen) are represented as spherical objects.
- (4) Electron Density: Represented by two isosurfaces; positive values are blue, negative values are yellow.
- (5) Vector Potential: Mapped to the floor of the calculation space (2D), positive values are red, negative values are blue, transparency increases as the value approaches 0.

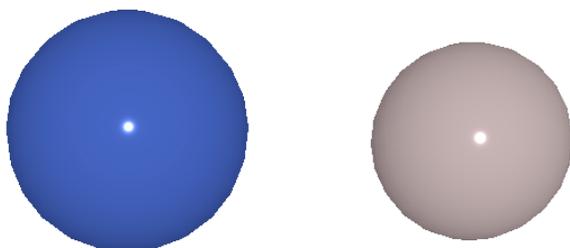
### Cube Data Rendering and Timeline

The app loads the following information from Cube files (\*.cube) on startup:

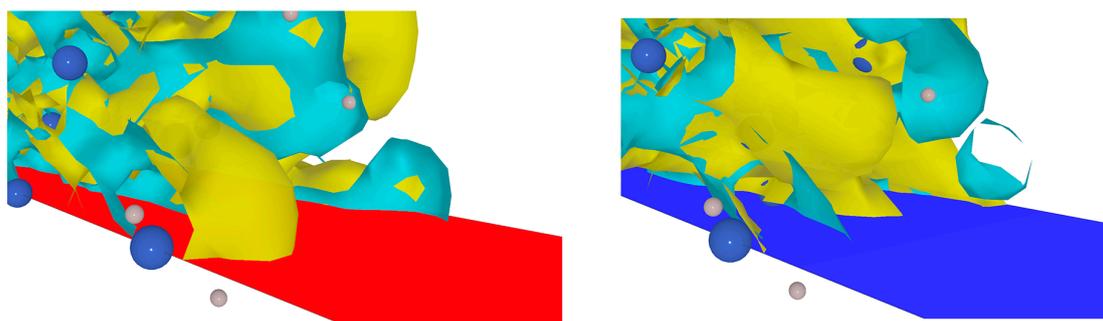
Information	Source
Change of Electron Density Data (all steps)	Files matching the *drho.cube pattern. Format is configurable.
Vector Potential Data (all steps)	Files matching the *Ac_x.cube, *Ac_y.cube, *Ac_z.cube patterns. Format is configurable.
Grid Parameters & Atomic Data	Header of the first loaded Cube file.
Timesteps	Six-digit number following "it" in filenames.

### Rendering Details:

- **Atoms:** Rendered as spheres with colors and sizes based on their type.



- **Electron Density:** The electron density is rendered on two isosurfaces (Isosurface), with integers in yellow and negatives in cyan. Isosurface is calculated using the Marching Cube method and the levels can be changed in the GUI panel described below.
- **Vector Potential:** The vector potentials are rendered as mapped planes on the floor of the computational space (the planes formed by the grids in the X and Z directions). In this case, the values taken by the Y-directional grid with fixed X and Z are averaged. The values of each grid are represented as red for integers and black for negatives. The closer the value is to 0, the more transparent the image becomes, with 0 being completely transparent. The values are normalized so that the maximum value is perfect red (#FF0000FFF) and black (#0000FFFFFF). As an example, the diagram below shows the results of rendering data when the vector potential is an integer (left) and a negative number (right).



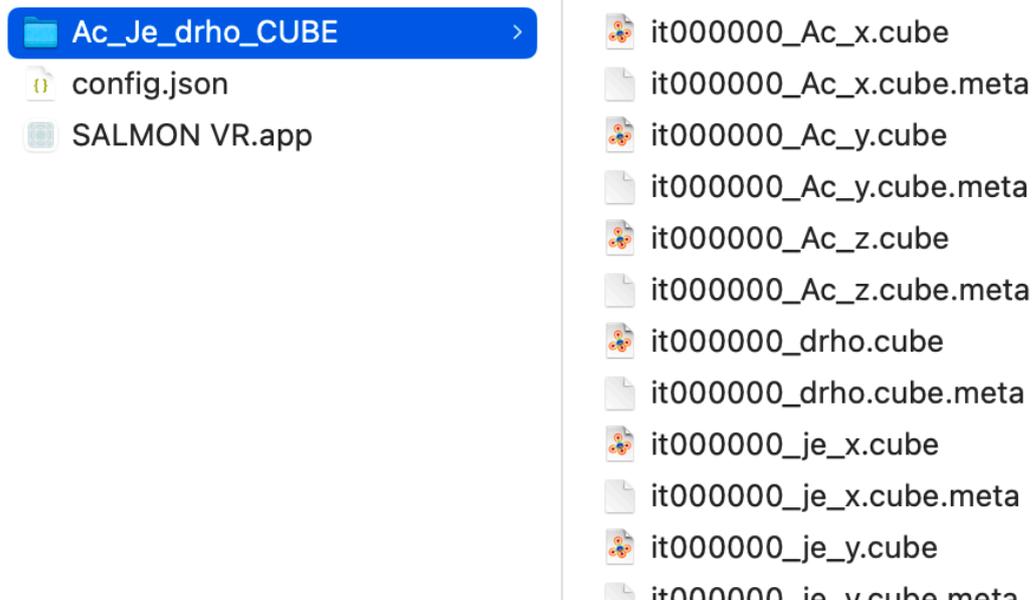
### Timeline and Timestep Interpolation

The app creates a timeline from loaded data. If timesteps are missing from the Cube file, values are linearly interpolated.

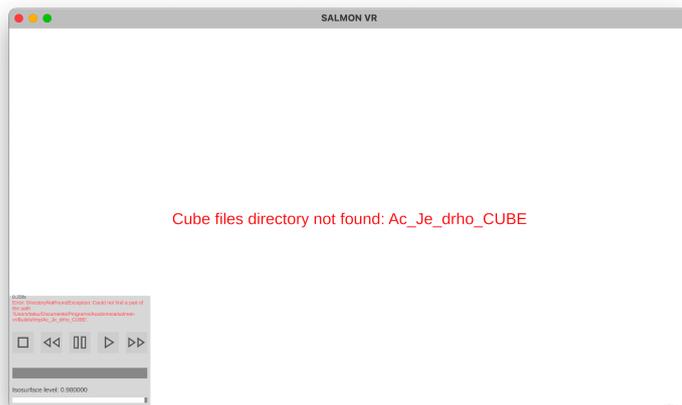
## Loading Cube Files

The app loads Cube files from the same directory as the app file.

For example, if the directory 'Ac\_Je\_drho\_CUBE', which contains the Cube file set, is the target, place it as shown in the following diagram in the Finder.



The target directory name can be changed from the settings file described below. If the following error message is displayed immediately after starting this application, the Cube file may not have been loaded correctly. Check that the directory location and name are correct.



### Configuration File (`config.json`)

You can modify app settings via `config.json` in the same directory as the app. Default values:

```
"CubeFilesDirectoryName": "Ac_Je_drho_CUBE",  
"InformationText": "0.25fs",
```

```
"TimestepPerSecond": 100,
"DrhoFileFormat": "*drho.cube",
"AcFileFormat": "*Ac_x.cube",
"EnableAcGammaCorrection": false,
"AcGamma": 2.2
```

The contents of each variable are as follows.

Key	Value	Type
CubeFilesDirectoryName	Name of the directory in which the set of Cube files to be loaded is located.	text
InfomationText	Additional text to display information on the GUI panel	text
TimestepPerSecond	Specifies the speed of the animation. The value specifies the amount of timesteps to advance per second when the timeline is regenerated (for 100: 100 timesteps per second).	integer
DrhoFileFormat	Electronic density file format.	text
AcFileFormat	Vector potential file format.	text
EnableAcGammaCorrection	Whether the transparency of the vector potential is gamma-corrected or not.	boolean
AcGamma	Coefficient of the above gamma correction (only valid if EnableAcGammaCorrection: true).	number

## Keyboard and Mouse Controls

The following keyboard and mouse combinations can be used to move the viewpoint freely in 3D space.

### Keyboard Control

Moving in space

Key	Action
w	Forwards
s	Backwards
a	Left
d	Right
e	Up

q	Down
---	------

Rotation of viewpoints

Key	Rotation
↑	Upwards
↓	Downwards
←	Left
→	Right

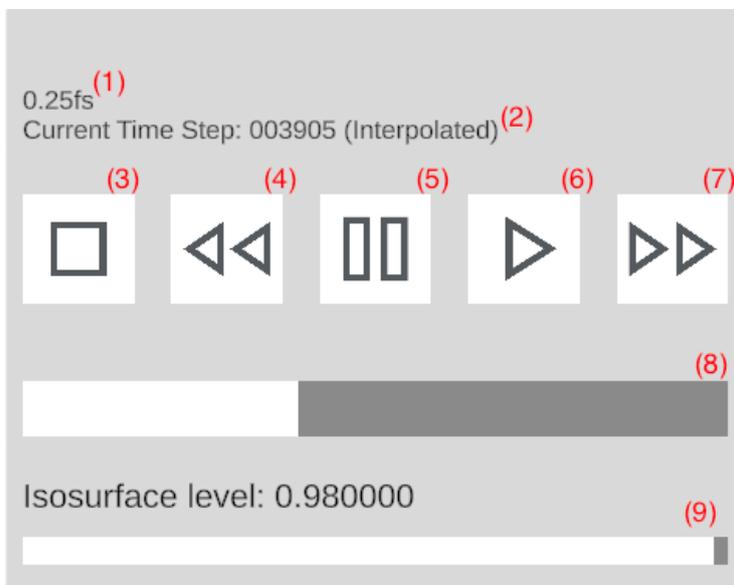
\*Simultaneous pressing of the Shift key: increased speed of movement and rotation.

Mouse Control

Operation	Action
Scroll	Forward and backward movement
Drag with the right button.	Rotation of viewpoints
Dragging the middle button	Vertical and horizontal movement

## GUI Panel Usage

The GUI panel allows for timeline control and isosurface level adjustment.



- (1) Additional information display (can be changed from the setting file)
- (2) Current time step value (Interpolated values are followed by Interpolated)
- (3) Stop
- (4) Double-speed reverse rplayback (the magnification increases each time it is pressed)
- (5) Stop temporarily.
- (6) Play

- (7) Double-speed regeneration (magnification is increased each time it is pressed)
- (8) Seek bar (position can be controlled by mouse dragging)
- (9) Change of Isosurface Level (value normalized to maximum value)

## Keyboard control of the timeline

The timeline can also be operated from keyboard input.

Key	Action
Space	Play/Stop
k	Play/Stop
j	Double-speed reverse (magnification increases with each input)
l	Double-speed (magnification increases with each input)

Frame-by-frame play

Key	Action
F	One time step forward.
b	Back 1 time step.

\*Variable every 10 time steps by pressing the Shift key simultaneously.