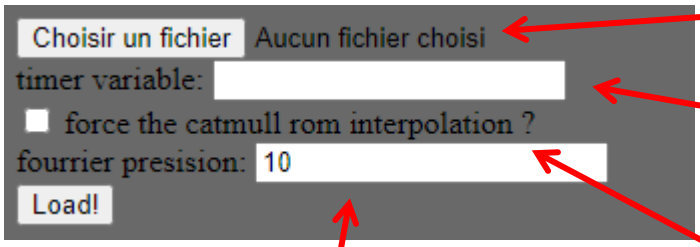


Select your animation file from blockbench

The timer variable is the variable use to know here you are in the animation (ex : age)

Force the program to use smooth interpolation, make the aproximation better. Is not check, the program will use the interpolation that was use in blockbench

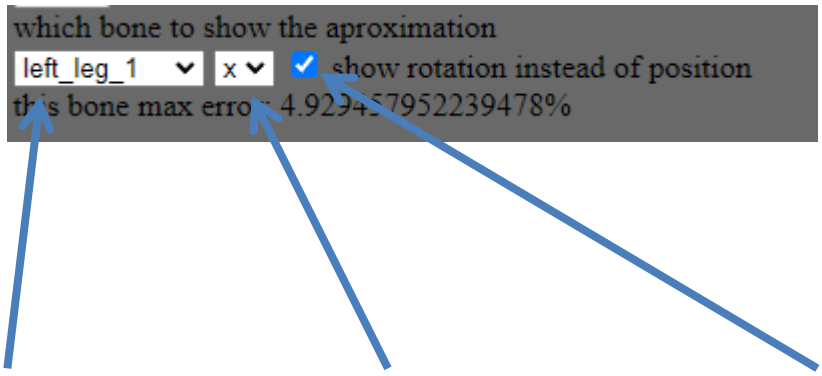
How many sin and cos will be use to describe the animation: fourier series are never exact and the more sin function you use, the closer to the real function you are.

Usualy 10 is a good value. You shouldn't need to go abouve 20 and don't excede 500 (integration error occur and you don't get the right value)

Then hit load!

Something like that will appear

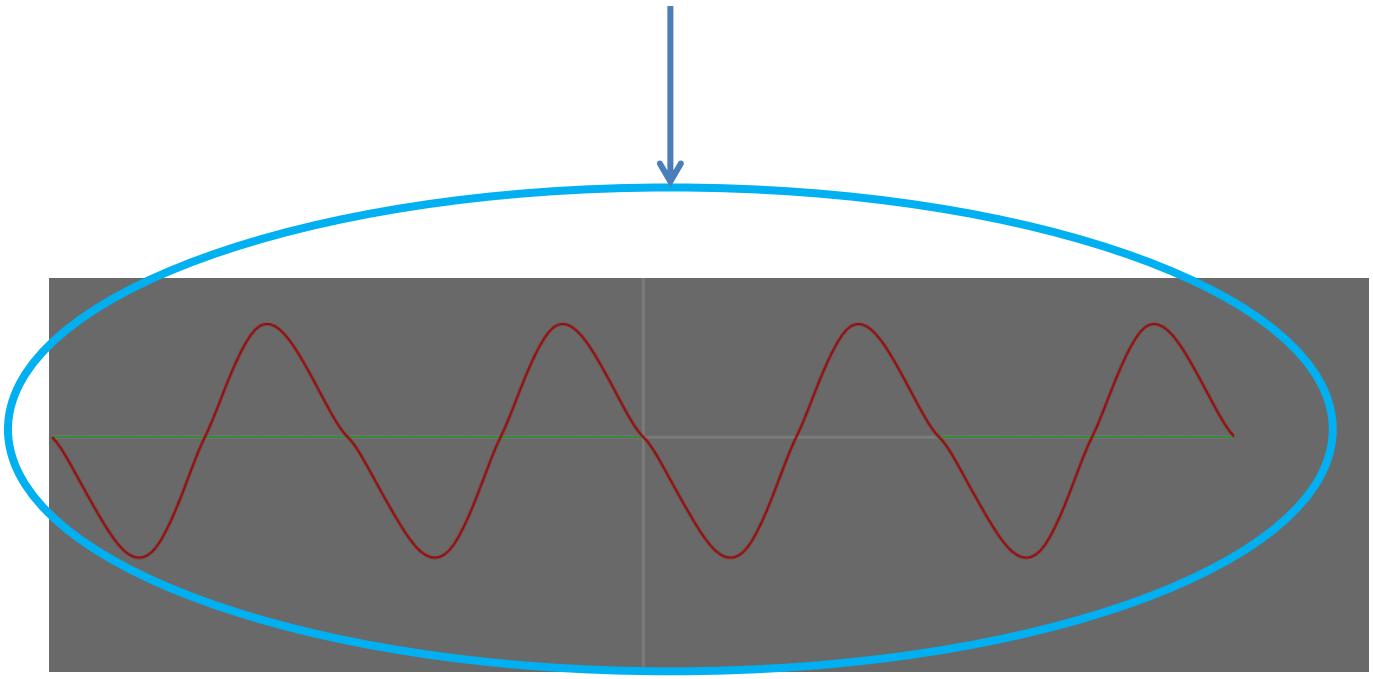
You can choose a bone to see a comparaison between the real animation and the aproximation

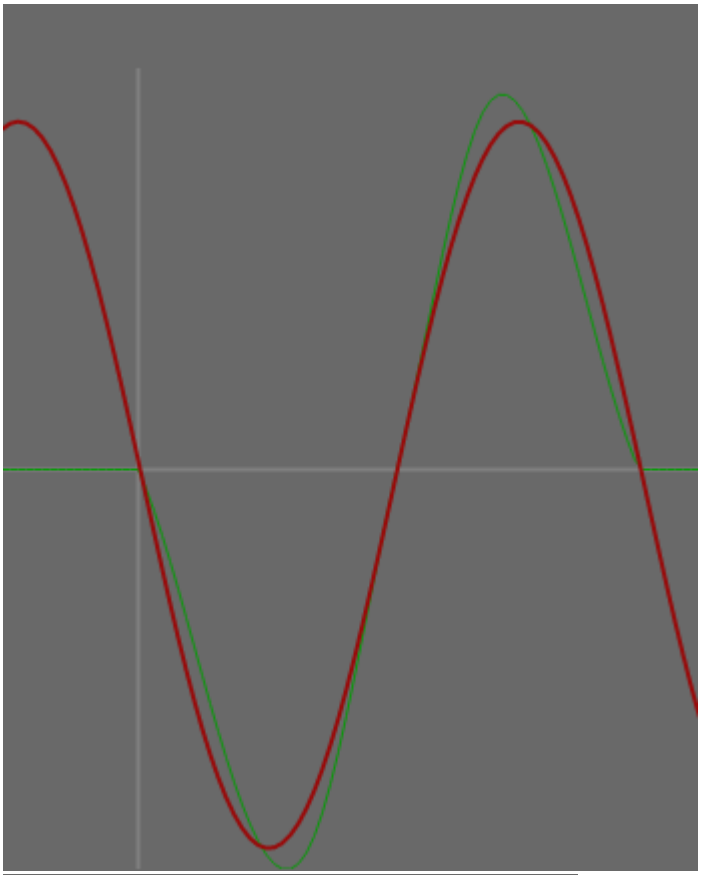


Select the bone, the axe and if you want the rotation or animation (checked = rotation, uncheck = position)

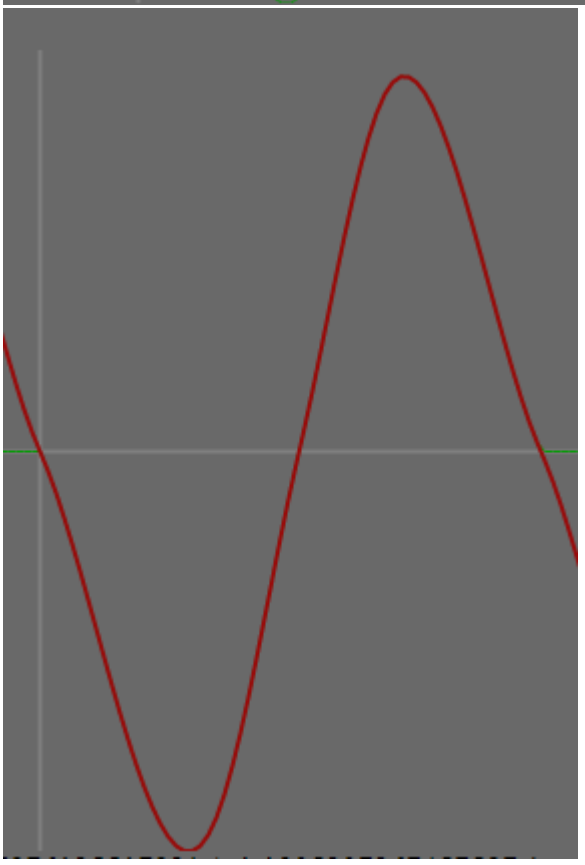
You can see the max error (how much the approximation is wrong): when you select a bone, you see the error on that bone. If you click ‘load’, you’ll get the worst on the model.

Preview of the animation function, in green you have the interpolation from keyframe, in red you have the fourier serie. If you don’t see the function, make sure you selected a valid combination of bone/axe/(position/rotation)





If you can see the green function here, you might need to increase the fourier presision but that will mean heavier fonction.



If you can’t see the green function, that mean the presision is enough.

Some info on the model.json format :

Because of how CPM 1.12 and blockbench work, incorporating the function in the model.json is not intuative but it is not hard.

The plug in that export your model to 1.12 can create up to 4 bone for a bone with a single bone in blockbench (here is an exple where in blockbench, flame_1 is just one bone with one cube.

```
{
    "id": "flame_1_cpm_dummy",
    "parent": "flame",
    "position": [3.75, 0, 0]
},
{
    "id": "flame_1",
    "parent": "flame_1_cpm_dummy",
    "rotation": [-15, 0, 0],
    "boxes": []
},
{
    "id": "flame_1_flame_1_wrapper_bone",
    "parent": "flame_1",
    "position": [-3.75, -10.5, -7.5],
    "boxes": []
},
{
    "id": "flame_1_flame_1_wrapper_bone_pivot_point",
    "parent": "flame_1_flame_1_wrapper_bone",
    "rotation": [0, 0, 0],
    "boxes": [
        {
            "textureOffset": [43, 22],
            "coordinates": [3.75, 8.5, 10.5, 0.001, 4, 3]
        }
    ]
},
```

We will use this field for position

And this one for rotation

The order of calculation (position and rotation) is not the same in blockbench and CPM 1.12. you can use the position and rotation in the same bone but be aware that this might lead to some issue with the bone not following the animation (all the more so if child bones are also animated).

If the _cpm_dummy is not present, you will have to create it if you have position animation.

Exemple with the X position

X

Y

Z

X

Y

Z

```
mirror: rotation : -13.778893173486479+2.3341098718557927 *cos(10.399697613551792*age)+13.472881176051919
*cos(20.799395227103584*age)-3.1038424822525763 *cos(31.199092840655375*age)-1.7156699729061402
*sin(10.399697613551792*age)+2.365143155602323 *sin(20.799395227103584*age)-1.2022248962872688 *sin(31.199092840655375*age)
,
+8.353683143014916-10.710771570052284 *cos(10.399697613551792*age)+3.313990996722047
*cos(20.799395227103584*age)-1.3730495477072389 *cos(31.199092840655375*age)-3.6718502867781755
*sin(10.399697613551792*age)+1.030284985978201 *sin(20.799395227103584*age)-0.3494451042037586 *sin(31.199092840655375*age)
,
+0
,
; position : -0.6759426942675322+0.9358356052047605 *cos(10.399697613551792*age)-0.18904194641270783
*cos(20.799395227103584*age)-0.13554695192372856 *cos(31.199092840655375*age)-13.34978600927487
*sin(10.399697613551792*age)+1.6754305848007491 *sin(20.799395227103584*age)+0.5915918879750661
*sin(31.199092840655375*age)
,
+6.486790035956601-9.538878976438841 *cos(10.399697613551792*age)+4.090075979920064
*cos(20.799395227103584*age)-1.5064318002780157 *cos(31.199092840655375*age)-4.099043188013166
*sin(10.399697613551792*age)+1.2055442729838481 *sin(20.799395227103584*age)-0.3875706560324486 *sin(31.199092840655375*age)
,
-2.7541707374917612+3.416064671433788 *cos(10.399697613551792*age)-1.0307878852802959
*cos(20.799395227103584*age)+0.5448723630125115 *cos(31.199092840655375*age)+7.846519358159276
*sin(10.399697613551792*age)-1.1786535960730096 *sin(20.799395227103584*age)-0.16985090536778527
*sin(31.199092840655375*age)
,
```

```
{
    "id": "mirror_cpm_dummy",
    "parent": "mirror_anim",
    "position" : [1, 2, -2]
},
{
    "id": "mirror",
    "parent": "mirror_cpm_dummy",
    "rotation": [0, 0, 7.5],
    "boxes": []
},
```

"1-0.6759426942675322+0.9358356052047605
*cos(10.399697613551792*age)-0.18904194641270783
*cos(20.799395227103584*age)-0.13554695192372856
*cos(31.199092840655375*age)-13.34978600927487
*sin(10.399697613551792*age)+1.6754305848007491
*sin(20.799395227103584*age)+0.5915918879750661
*sin(31.199092840655375*age)"