HEPAComp v1.6

Nikolay Ikonomov

October 20, 2024

Contents

1 Install on Windows **3**

2 Install on Linux **3**

3 Computational options for PARI/GP **4**

4 Plotting options for gnuplot **5**

5 Errors **6**

6 Mathematical background **6**
6.1 Padé approximant of $f(z)$ 6
6.2 Hermite-Padé approximant of $[1, f, g]$ type I 7
6.3 Hermite-Padé approximant of $[1, f, g, h]$ type I 7
6.4 Two-point Padé approximant of f, g 7
6.5 Discriminant for Hermite-Padé approximant of $[1, f, g]$ 8
6.6 Special index for Hermite-Padé approximant of $[1, f, g]$ type II 8
6.7 Special index for Hermite-Padé approximant of $[1, f, g, h]$ type II 8
6.8 Discriminant for Hermite-Padé approximant of $[1, f, g, h]$ 9

7 Changelog **9**

8 History **10**

9 Download **11**

10 External vector file **11**

This software application computes a rational approximant to a function, given by its Taylor expansion at the origin. Then the zeros of the polynomials of the rational approximant are plotted on the complex plane. This is achieved by using two applications, `PARI/GP` and `gnuplot`.

This application computes the following approximants by using `PARI/GP`:

- PADE – Padé Approximant of a function $F(z)$;
- HEPA – Hermite-Padé Approximant of $[1, F, G]$ Type I;
- MHEPA – Hermite-Padé Approximant of $[1, F, G, H]$ Type I;
- TPPA – Two-point Padé Approximant of F and G;
- HEPAD – Discriminant for Hermite-Padé Approximant of $[1, F, G]$;
- HEPAS – Determinant 2x2 for Hermite-Padé Approximant of $[1, F, G]$ Type II;
- MHEPAS – Determinant 2x2 and 3x3 for Hermite-Padé Approximant of $[1, F, G, H]$ Type II;
- MHEPAD – Discriminant for Hermite-Padé Approximant of $[1, F, G, H]$.

The result from the calculations are number of polynomials. The application finds the zeros of the polynomials, and plots these zeros on the complex plane with a certain color:

- PADE – two polynomials: blue, red;
- HEPA – three polynomials: blue, red, black;
- MHEPA – four polynomials: blue, red, black, green;
- TPPA – two polynomials: blue, red;
- HEPAD – one polynomial: magenta;
- HEPAS – one polynomial: cyan;
- MHEPAS – two polynomials: cyan, orange;
- MHEPAD – one polynomial: magenta.

1 Install on Windows

HEPAComp is ready to go on Windows XP 32-bit, Windows 7 64-bit, Windows 10 64-bit.

The application consists of several parts:

- `PARI/GP` – computer algebra system – in folder `./pari/`
- `gnuplot` – plotting utility – in folder `./gnuplot/`
- `SumatraPDF` – PDF/EPS viewer – in folder `./sumatra/`
- `HEPAComp.jar` – user interface written in Java

Install Oracle Java – JDK LTS:
`https://www.oracle.com/java/technologies/downloads/`

The application should work on all versions of Java 7 and above. In case of problems with the user interface, please install OpenJDK 17 from the links above.

After installation, start `HEPAComp.jar` and the user interface should appear. Press Start, wait a while, then press Create plot. The result should be displayed in a window.

Application paths can be changed from `config.ini`. Software applications mentioned above may be replaced by newer versions and then input proper path in `config.ini`.

Windows XP 32-bit requires editing `config.ini`, so that PARI/GP 32-bit is loaded. Restart HEPAComp, since configuration is read only on startup.

2 Install on Linux

HEPAComp is almost ready to go on Linux.

- `PARI/GP` – 64-bit executables are provided, along with the official sources and a script for compilation, it takes about 5 minutes
- `gnuplot` – install from the package manager, i.e. `apt install gnuplot-nox`
- `PDF viewer` – edit `config.ini` and select between atril/evince/okular or type the path
- `HEPAComp.jar` – edit `HEPAComp.sh` and launch it

Old versions of Java – 7, 8 – may require this option for anti-aliased fonts:
`java -Dawt.useSystemAAFontSettings=on -jar HEPAComp.jar`

New versions of Java – 11, 17 – may require this option for HiDPI displays:
`java -Dsun.java2d.uiScale=2.0 -jar HEPAComp.jar`

Options can be specified by export:

```
export _JAVA_OPTIONS="-Dawt.useSystemAAFontSettings=on"
java -jar HEPAComp.jar
```

HEPAComp.jar is written in Java Swing. For completeness, JavaFX on HiDPI displays:
`java -Dglass.gtk.uiScale=2.0 -jar myapp.jar`

3 Computational options for PARI/GP

Press Start once and wait for the text "Completed in". Then press Create plot. Select different options for the plot and press Create plot again. If the computation does not start, edit `config.ini` and change the version of `PARI/GP`.

All files are placed in the subdirectory `./output/`. Open the log file to see information about the current session.

Session name: Determined automatically by the type of the approximant, the degree of the polynomials and four random digits. Select the checkbox to input the name. The name can contain digits 0-9, letters A-z, a-z, underscore, dash and round brackets ().

The session name is checked against all `*.log` files in the `output` directory. The name should be unique, otherwise the data files may contain several sets of points, since `PARI/GP` appends the information in the files.

Precision (digits): Significant decimal digits after the decimal point. This must be an integer number, at least 1. By default this is set to 1000. Incresing this number increases somewhat the memory used and increases a lot the computational time.

Polynomial degree: "Diagonal" approximation: All polynomials (which are the result of the approximation) have this degree. Denote the degree by N. Only the first polynomial, named $Q_0(z)$, may have $N-1$ or N zeros. The rest of the polynomials are guaranteed to have exactly N zeros. The degree must be an integer value, at least 1. By default this is set to 30. Increasing this number significantly increases the memory and computation time.

"Non-diagonal" approximation: input several integer values separated by comma. Their count must be equal to the approximant's number of polynomials. PADE, TPPA, TPPB have by 2 polynomials; HEPA, HEPAD, HEPAS – 3; MHEPA, MHEPAS, MHEPAD – 4.

Multiple approximations: Select the checkbox to set an integer number, greater than the initial degree, to compute several approximations. This only works with "diagonal" approximation. Number of 50 means that approximants for $30, 31, 32, \ldots 50$ are computed. By default the checkbox is not selected.

Taylor expansion: Maximum order of the Taylor expansion. The formula is: add the degrees of the polynomials, the number of the polynomials and subtract 1. For "diagonal" approximation the formula is: $pol * N + pol - 1$. For example, let $N = 30$. PADE has 2 polynomials, then the maximal order is 61. HEPA has 3 polynomials, order is 92. MHEPA has 4 polynomials, order is 123.

Input this number only for the error "polcoeff: non existent component in truecoeff". This happens when the function is divided by z, z^2, z^3. Increase the number by 1, 2, 3, respectively.

Approximant:

- PADE – Padé Approximant of $F(z)$,
- HEPA – Hermite-Padé Approximant of $[1, F, G]$ Type I,
- MHEPA – Hermite-Padé Approximant of $[1, F, G, H]$ Type I,
- TPPA – Two-point Padé Approximant of F, G with $/N+1, N/$,
- TPPB – Two-point Padé Approximant of F, G with $/N, N+1/$,
- HEPAD – HEPA Discriminant,

- HEPAS – HEPA Determinant 2x2 Type II,
- MHEPAS – MHEPA Determinant 2x2 and 3x3 Type II,
- MHEPAD – MHEPA Discriminant.

Algorithm:

- Matrix method – the default, any new algorithm is compared with the matrix method,
- Viskovatov algorithm – new algorithm, which extends the classical Viskovatov algorithm for Padé approximant.

The functions: Three functions can be typed in or selected by the dropdown menu. Their internal names are funF, funG and funH. Note that funF is evaluated first, then funG and finally funH. `PARI/GP` uses real numbers if there is at least one real number in the function. Use capital letter I for the imaginary unit. Use `^(1/2)` for square root and `^(1/3)` for cube root. Use exp, log, sin, cos, tan, cotan for the transcendental functions. Reference the previous functions by funF and funG, like `funF^2`, `funG^(-2/3)`, etc.

After calculations: The approximant is calculated at the origin. Select between staying at the origin and sending to the infinity point.

Memory (in gigabytes): Increase this for the error "the PARI stack overflows". By default this is set to 2.

Start: When the button is pressed, all data from the panel is collected and a script named `HEPA_x_30_1000_8329.gp` is created. Then `PARI/GP` reads this script, makes the computations, finds the zeros of the polynomials, and saves them as `HEPA_x_30_1000_8329q0.gpd`, `HEPA_x_30_1000_8329q1.gpd` and `HEPA_x_30_1000_8329q2.gpd` for $Q_0(z)$, $Q_1(z)$, and $Q_2(z)$ respectively. These files have to be parsed for `gnuplot`, therefore the final data files are saved as `*.gpt`.

Note that `_x_` denotes the matrix method, while `_v_` denotes the Viskovatov algorithm. Points are saved in files ending with `q0 q1 q2 q3` for all approximants. See page 2 for the number of files for each approximant.

Stop: Stop the running process. If needed, press it to make the Start button active.

4 Plotting options for gnuplot

List of sessions: When the Start button is clicked, the session name is added on top of this list. This happens even if the approximant cannot be computed. In that case, click on the session name and select Rem to remove the session. Alternatively, use the right mouse button on the selected session.

Each new session in inserted on top of the list, so that when all sessions are plotted, the session on top of the list will be also on top of all other plots in the PDF file. Change the behaviour by using the Top/Bot buttons.

Multiple sessions can be added or deleted at once: hold the Ctrl or Shift key to select several sessions. This applies both to the file chooser and the list of sessions.

Select a session with the mouse by left-clicking on it. Then press the Polynomials button, a small window should appear, it allows choosing which polynomials to plot. The polynomials are referenced by 0,1,2,3 in the session name, and by Q0,Q1,Q2,Q3 in the window.

Polynomials can also be chosen by double-clicking with the mouse on the session name, or by right-clicking on a session name, and using the menu that appears.

Range of the plot: The range for the X and Y axes can be automatically determined by `gnuplot` or manually entered. Some common ranges are there for convenience.

Range for X and Y axis: The values can be integer and real numbers.

File type: Choose between Portable Document Format and Encapsulated PostScript.

Create plot: This button creates a plot from all available sessions and creates a PDF file with current date and time `Result_20201015_142014.pdf`. Please accept apologies for this inconvenience, but the plot file cannot be named after the sessions, since multiple sessions can be plotted at once, this may lead to very long file names.

Finally, the file viewer opens the PDF file and the result is displayed.

5 Errors

Note that the application works with `PARI/GP` versions 2.5.5 and 2.7.6. Version 2.5.5 is 32-bit and was the default until v1.0. From `HEPAComp` v1.0 the default is `PARI/GP` v2.7.6 64-bit, which does not have memory limits.

The latest stable `PARI/GP` v2.11.4 64-bit is also provided, but it is not tested. I am simply fond of v2.7.6 permanently, you can experiment with newer versions if you like.

division by zero: Change the functions and carefully check the input values.

3 should divide valuation (= 1) in sqrtn: This happens when some part of a function does not have a Taylor expansion at all. For example $z^{1/3}f(z)$, even if $f(z)$ has a Taylor expansion.

polcoeff: non existent component in truecoeff: This happens when a function is divided by z, z^2, z^3, etc. Increase the Taylor expansion by 1, 2, 3, respectively.

matsolve: the PARI stack overflows: Allocate more than 2 GB of memory.

matsolve: not enough memory: The internal memory limit for `PARI/GP` 32-bit is around 1500 MB. Edit `config.ini` and use `PARI/GP` 64-bit, which does not have memory limits.

gnuplot warnings: When all points are on the real axis, `gnuplot` automatically adjusts the imaginary axis to $[-1, 1]$. When a plot file is missing, `gnuplot` skips it.

6 Mathematical background

This section explains briefly the mathematical reasoning behind the approximants. Detailed explanations are in `./docs/algorithms/appr_algorithms.pdf`

6.1 Padé approximant of $f(z)$

Let the function $f(z)$ be given and its Taylor expansion be:

$$f(z) = \sum_{j=0}^{\infty} c_j z^j.$$

Let the two polynomials $Q_{n,0}$, $Q_{n,1} \in \Pi_n$, have no common divisor, $Q_{n,1} \neq 0$, and satisfy the condition:

$$Q_{n,1}(z)f(z) - Q_{n,0}(z) = \mathcal{O}\left(z^{2n+1}\right).$$

We set $Q_{n,0}/Q_{n,1}$ for the diagonal Padé approximant of $f(z)$.

6.2 Hermite-Padé approximant of $[1, f, g]$ type I

Let the functions $f(z)$ and $g(z)$ be given and their Taylor expansions be:

$$f(z) = \sum_{j=0}^{\infty} c_j z^j, \quad g(z) = \sum_{j=0}^{\infty} d_j z^j.$$

Let the polynomials $Q_{n,0}$, $Q_{n,1}$, $Q_{n,2} \in \Pi_n$, have no common divisor and satisfy the condition:

$$(Q_{n,0} + Q_{n,1} f + Q_{n,2} g)(z) = \mathcal{O}\left(z^{3n+2}\right).$$

Define $[a_1, a_2, a_3]$ as a "collection" of three objects. We set $[Q_{n,0}, Q_{n,1}, Q_{n,2}]$ for the Hermite-Padé polynomials for the collection of the three functions $[1, f, g]$.

6.3 Hermite-Padé approximant of $[1, f, g, h]$ type I

Let the functions $f(z)$, $g(z)$ and $h(z)$ be given and their Taylor expansions be:

$$f(z) = \sum_{j=0}^{\infty} c_j z^j, \quad g(z) = \sum_{j=0}^{\infty} d_j z^j, \quad h(z) = \sum_{j=0}^{\infty} e_j z^j.$$

Let the polynomials $Q_{n,0}$, $Q_{n,1}$, $Q_{n,2}$, $Q_{n,3} \in \Pi_n$, have no common divisor and satisfy the condition:

$$(Q_{n,0} + Q_{n,1} f + Q_{n,2} g + Q_{n,3} h)(z) = \mathcal{O}\left(z^{4n+3}\right).$$

Define $[a_1, a_2, a_3, a_4]$ as a "collection" of four objects. We set $[Q_{n,0}, Q_{n,1}, Q_{n,2}, Q_{n,3}]$ for the Hermite-Padé polynomials for the collection of the four functions $[1, f, g, h]$.

6.4 Two-point Padé approximant of f, g

Let the function $f(z)$ be given, with a Taylor expansion around zero:

$$f(z) = \sum_{j=0}^{\infty} c_j z^j,$$

and the function $g(z)$ with a Taylor expansion around infinity:

$$g(z) = \sum_{j=0}^{\infty} \frac{d_j}{z^j}.$$

Let the two polynomials $Q_{n,0}$, $Q_{n,1} \in \Pi_n$ have no common divisor, $Q_{n,1} \neq 0$, and satisfy the conditions:

$$\begin{aligned} Q_{n,1}(z) f(z) - Q_{n,0}(z) &= \mathcal{O}(z^{n+1}), \\ Q^{\infty}_{n,1}(z) g(z) - Q^{\infty}_{n,0}(z) &= \mathcal{O}\left(\frac{1}{z^n}\right), \end{aligned}$$

where $Q^{\infty}_{n,0}(z) = Q_{n,0}(1/z)$, $Q^{\infty}_{n,1}(z) = Q_{n,1}(1/z)$. We refer to this case as "case $/n+1, n/$". Alternatively,

$$\begin{aligned} Q_{n,1}(z) f(z) - Q_{n,0}(z) &= \mathcal{O}(z^{n}), \\ Q^{\infty}_{n,1}(z) g(z) - Q^{\infty}_{n,0}(z) &= \mathcal{O}\left(\frac{1}{z^{n+1}}\right), \end{aligned}$$

and refer to it as "case $/n, n+1/$". We set $Q_{n,0}/Q_{n,1}$ as the two-point Padé approximant at a finite point (including zero), and set $Q^{\infty}_{n,0}/Q^{\infty}_{n,1}$ as the two-point Padé approximant at the infinity point.

6.5 Discriminant for Hermite-Padé approximant of $[1, f, g]$

Special case of Hermite-Padé approximant of $[1, f, g]$. Recall that

$$(Q_{n,0} + Q_{n,1}f + Q_{n,2}g)(z) = \mathcal{O}\left(z^{3n+2}\right).$$

Find the zeros of the polynomial

$$(Q_{n,1})^2 - 4Q_{n,0}Q_{n,2}.$$

6.6 Special index for Hermite-Padé approximant of $[1, f, g]$ type II

Special case of Hermite-Padé approximant of $[1, f, g]$. Recall that

$$(Q_{n,0} + Q_{n,1}f + Q_{n,2}g)(z) = \mathcal{O}\left(z^{3n+2}\right).$$

Compute two consecutive indexes $(n-1, n-1, n-1)$ and (n, n, n), then make a determinant from the polynomials corresponding to the functions f and g:

$$\begin{vmatrix} Q_{n-1,1} & Q_{n-1,2} \\ Q_{n,1} & Q_{n,2} \end{vmatrix}$$

This is equivalent to computing these two indexes:

$$\begin{matrix} (n, n, n-1) \\ (n, n-1, n) \end{matrix}$$

and then computing the following determinant:

$$\begin{vmatrix} Q_{n,1} & Q_{n-1,2} \\ Q_{n-1,1} & Q_{n,2} \end{vmatrix}$$

Expand the determinant and find the zeros of the resulting polynomial.

6.7 Special index for Hermite-Padé approximant of $[1, f, g, h]$ type II

Special case of Hermite-Padé approximant of $[1, f, g, h]$. Recall that

$$(Q_{n,0} + Q_{n,1}f + Q_{n,2}g + Q_{n,3}h)(z) = \mathcal{O}\left(z^{4n+3}\right).$$

Compute three consecutive indexes $(n-2, n-2, n-2, n-2)$, $(n-1, n-1, n-1, n-1)$ and (n, n, n, n), then make two determinants: 2x2 corresponding to the functions g and h:

$$\begin{vmatrix} Q_{n-1,2} & Q_{n-1,3} \\ Q_{n,2} & Q_{n,3} \end{vmatrix}$$

and 3x3 corresponding to the functions f, g and h:

$$\begin{vmatrix} Q_{n-2,1} & Q_{n-2,2} & Q_{n-2,3} \\ Q_{n-1,1} & Q_{n-1,2} & Q_{n-1,3} \\ Q_{n,1} & Q_{n,2} & Q_{n,3} \end{vmatrix}$$

This is equivalent to computing these three indexes:

$$\begin{matrix} (n, n, n-1, n-1) \\ (n, n-1, n, n-1) \\ (n, n-1, n-1, n) \end{matrix}$$

and then computing the following determinants:

$$\begin{vmatrix} Q_{n,2} & Q_{n-1,3} \\ Q_{n-1,2} & Q_{n,3} \end{vmatrix}$$

and

$$\begin{vmatrix} Q_{n,1} & Q_{n-1,2} & Q_{n-1,3} \\ Q_{n-1,1} & Q_{n,2} & Q_{n-1,3} \\ Q_{n-1,1} & Q_{n-1,2} & Q_{n,3} \end{vmatrix}$$

Expand the determinants and find the zeros of the two polynomials.

6.8 Discriminant for Hermite-Padé approximant of $[1, f, g, h]$

Special case of Hermite-Padé approximant of $[1, f, g, h]$. Recall that

$$(Q_{n,0} + Q_{n,1}f + Q_{n,2}g + Q_{n,3}h)(z) = \mathcal{O}\left(z^{4n+3}\right).$$

Find the zeros of the polynomial

$$18Q_{n,0}Q_{n,1}Q_{n,2}Q_{n,3} - 4Q_{n,0}(Q_{n,2})^3 + (Q_{n,1})^2(Q_{n,2})^2 - 4(Q_{n,1})^3Q_{n,3} - 27(Q_{n,0})^2(Q_{n,3})^2.$$

7 Changelog

- 1.6 (October 20, 2024)
  - Added MHEPAD (Matrix/Viskovatov)
- 1.5 (February 5, 2023)
  - Allow loading external vector file instead of Taylor expansion for the functions
- 1.4 (May 24, 2021)
  - Added new Viskovatov algorithm for MHEPA, MHEPAS
- 1.3 (October 15, 2020)
  - Added new Viskovatov algorithm for PADE, HEPA, HEPAD, HEPAS
  - Redesigned the user interface to allow panel resize
- 1.2 (July 30, 2017)
  - Changed main equation to determinant 2x2 and 3x3 in HEPAS and MHEPAS. The transformation $z \to 1/z$ is now applied after the determinant is computed. This way the two approaches described above give equivalent results.
- 1.1 (May 28, 2017)
  - Added approximant Special index for MHEPA (MHEPAS)
- 1.0.0 (March 20, 2016)
  - Added list of sessions, allows plotting of several approximants at once
  - Restored context menu, added context menu for the list
  - Removed the fork MPPAComp, added approximant TPPA in HEPAComp
  - Added approximant Discriminant for HEPA (HEPAD)

  - Added approximant Special index for HEPA (HEPAS)

- 0.9.9 (December 31, 2014)
  - PARI/GP Scripts cleaned up and proofed
  - Removed approximant AHEPA (no functions applicable)
  - Removed menus, context menu (code cleanup)
  - Program is now cross-platform, see config.ini

- 0.9.7 (April 6, 2014)
  - Forked the program as MPPAComp
  - Added approximant TPPA in MPPAComp

- 0.9.5 (February 23, 2014)
  - Changed to matsolve in the algorithms, 50% faster than before
  - Added memory option for PARI/GP, matsolve needs a lot of memory

- 0.9 (February 18, 2014)
  - Added approximant AHEPA
  - Added list of working configurations
  - Added menus, context menu
  - Interface redesigned

- 0.7 (January 9, 2014)
  - Added approximant MHEPA
  - Interface streamlined

- 0.5 (December 15, 2013)
  - Initial version
  - Added approximant PADE
  - Added approximant HEPA

8 History

The most important aspects of the evolution of the algorithms:

- October 13, 2024 – added MHEPAD (Matrix/Viskovatov)
- May 23, 2021 – added Viskovatov algorithm for MHEPAS
- April 25, 2021 – added Viskovatov algorithm for MHEPA
- October 14, 2020 – added Viskovatov algorithm for HEPAD and HEPAS
- May 15, 2020 – added Viskovatov algorithm for PADE and HEPA
- July 29, 2017 – changed equation to determinant in HEPAS and MHEPAS
- May 28, 2017 – added MHEPAS

- March 19, 2016 – added HEPAD and HEPAS
- December 28, 2014 – added "non-diagonal" version for PADE, HEPA, MHEPA
- April 2, 2014 – modified for TPPA
- February 23, 2014 – changed from manual matrix reduction to matsolve(), in effect removed the algorithm from February 23, 2013
- February 4, 2014 – polynomial size from polroots(): changed from fixed length to dynamic length with length(pts)
- January 30, 2014 – added polroots() for AHEPA
- January 27, 2014 – modified for AHEPA
- January 6, 2014 – modified for MHEPA
- December 3, 2013 – improvements for arrA, arrB
- November 27, 2013 – modified for Hermite-Padé
- November 22, 2013 – added polroots()
- February 23, 2013 – initial version

9 Download

Download the applications:

- PARI/GP v2.7.6 `http://pari.math.u-bordeaux.fr/`
- gnuplot v4.6.5 `https://gnuplot.sourceforge.net/`
- SumatraPDF v3.1.2 `https://www.sumatrapdfreader.org/`
- HEPAComp v1.6 `https://justmathbg.info/hepacomp.html`

10 External vector file

This version of HEPAComp can load external vector files instead of Taylor expansion for the functions. Switch `funF` to `arrC`, then right-click on the text field and select External. Choose a vector file and press Start as usual.

PARI/GP accepts normal text files with a single real/complex value on a single line.

```
1.23
2.34 E1
2.34E+1
-5.1004e-5
-5.1004 E-5 + 4.2005 E-4*I
```

Spaces do not matter anywhere on the line. There must be a sign for the imaginary part. This also works: `-5.2+-4.3*I` and equals $-5.2 - 4.3i$. In contrast: `-5.2 4.3*I` equals -5.24 and warning is displayed for unused characters `.3*I`.

The external vector file is saved again by PARI/GP for verification – ends in `_arrC.txt` `_arrD.txt` `_arrE.txt` – compare with the original files.

See `./docs/vector.txt` for example input.