HEPAComp v1.7

Nikolay Ikonomov

7 July 2025

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 HEPAComp on Windows

HEPAComp is ready for use on Windows XP 32-bit, Windows 7 64-bit, Windows 10 64-bit. Extract the archive somewhere and launch HEPAComp.jar or HEPAComp.exe. The application contains several parts:

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

The user interface written in Java should work on all versions of Java 7 and above. HiDPI displays may require Java 11 and later. Recommended is Oracle Java LTS 11, 17, 21:
https://www.oracle.com/java/technologies/downloads/

Start `HEPAComp` 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 paths 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 HEPAComp on Linux

HEPAComp requires installation of packages on Linux.

- `PARI/GP` – 64-bit executables are in `./pari/` compiled on Debian 12, along with the official sources and a script for compilation, compiles in 3 minutes; or `apt install pari-gp`, but it pulls TeXLive and others
- `gnuplot` – install from the package manager: `apt install gnuplot-nox`
- `fileview` – edit `config.ini` and select atril/evince/okular/other
- `HEPAComp.jar` – install Java from the package manager: `apt install openjdk-17-jdk` and edit `HEPAComp.sh` and launch it
- `HEPAComp.qt6` – Qt version may need additional libraries: `ldd HEPAComp.qt6` and then apt install libxcb1 libxcb-cursor0 libxcb-image0 libxcb-keysyms1 libxcb-icccm4 libpcre2-16-0 and launch it

HiDPI displays (Java accepts integers, Qt accepts reals):
Java 7, 8: `java -Dawt.useSystemAAFontSettings=on -jar HEPAComp.jar`
Java 11, 17, 21: `java -Dsun.java2d.uiScale=2 -jar HEPAComp.jar`
Qt 6.8 on Linux: `QT_SCALE_FACTOR=2.25 ./HEPAComp.qt6`
Qt 6.8 on Windows: `set QT_SCALE_FACTOR=1.5 && HEPAComp.exe`

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 5 gigabytes of RAM.

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 Messages and others

HEPAComp has been fully tested 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.

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.

SumatraPDF does not open EPS files. In general, viewing EPS files require GhostScript to be installed, and the provided version of SumatraPDF works only with GhostScript ≤ 9.53. Use SumatraPDF ≥ 3.3 with GhostScript ≥ 9.54 to view EPS files.

6 External vector file

Starting from HEPAComp v1.5 there is the option to 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 file. 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.

7 Mathematical background

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

7.1 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 $f(z)$.

7.2 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 $[1, f, g]$.

7.3 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 $[1, f, g, h]$.

7.4 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:

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

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,

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

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.

7.5 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

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

7.6 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:

$$(n, n, n-1)$$
$$(n, n-1, n)$$

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.

7.7 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.

7.8 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

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

8 Changelog

- Version 1.7 (7 July 2025)
  - Allowed Viskovatov algorithm and multiple approximations
  - Added new user interface on Qt 6.8 [Qt]
  - Refactored source code [Java]
- Version 1.6 (20 October 2024)
  - Added MHEPAD (Matrix/Viskovatov)
- Version 1.5 (5 February 2023)
  - Allow loading external vector file instead of Taylor expansion for the functions
- Version 1.4 (24 May 2021)
  - Added new Viskovatov algorithm for MHEPA, MHEPAS
- Version 1.3 (15 October 2020)
  - Added new Viskovatov algorithm for PADE, HEPA, HEPAD, HEPAS
  - Redesigned the user interface to allow panel resize
- Version 1.2 (30 July 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.
- Version 1.1 (28 May 2017)
  - Added approximant Special index for MHEPA (MHEPAS)
- Version 1.0.0 (20 March 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)
- Version 0.9.9 (31 December 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
- Version 0.9.7 (6 April 2014)
  - Forked the program as MPPAComp
  - Added approximant TPPA in MPPAComp

- Version 0.9.5 (23 February 2014)
  - Changed to matsolve in the algorithms, 50% faster than before
  - Added memory option for PARI/GP, matsolve needs a lot of memory
- Version 0.9 (18 February 2014)
  - Added approximant AHEPA
  - Added list of working configurations
  - Added menus, context menu
  - Interface redesigned
- Version 0.7 (9 January 2014)
  - Added approximant MHEPA
  - Interface streamlined
- Version 0.5 (15 December 2013)
  - Initial version
  - Added approximant PADE
  - Added approximant HEPA

9 History

The most important aspects of the evolution of the algorithms:

- 13 October 2024 – added MHEPAD (Matrix/Viskovatov) [v1.6]
- 23 May 2021 – added Viskovatov algorithm for MHEPAS [v1.4]
- 25 April 2021 – added Viskovatov algorithm for MHEPA
- 14 October 2020 – added Viskovatov algorithm for HEPAD and HEPAS [v1.3]
- 15 May 2020 – added Viskovatov algorithm for PADE and HEPA
- 29 July 2017 – changed equation to determinant in HEPAS and MHEPAS [v1.2]
- 28 May 2017 – added MHEPAS [v1.1]
- 19 March 2016 – added HEPAD and HEPAS [v1.0.0]
- 28 December 2014 – added "non-diagonal" version for PADE, HEPA, MHEPA [v0.9.9]
- 2 April 2014 – modified for TPPA [v0.9.7]
- 23 February 2014 – changed from manual matrix reduction to matsolve(), in effect removed the algorithm from 23 February 2013 [v0.9.5]
- 4 February 2014 – polynomial size from polroots(): changed from fixed length to dynamic length with length(pts) [v0.9]
- 30 January 2014 – added polroots() for AHEPA
- 27 January 2014 – modified for AHEPA

- 6 January 2014 – modified for MHEPA [v0.7]
- 3 December 2013 – improvements for arrA, arrB [v0.5]
- 27 November 2013 – modified for Hermite-Padé
- 22 November 2013 – added polroots()
- 23 February 2013 – initial version

10 Credits

Software applications used by HEPAComp for Windows:

- PARI/GP – computer algebra system – v.2.7.6
  https://pari.math.u-bordeaux.fr/
  https://pari.math.u-bordeaux.fr/pub/pari/windows/OLD/2.7/
- gnuplot – plotting utility – v.4.6.5
  https://gnuplot.sourceforge.net/
  https://sourceforge.net/projects/gnuplot/files/gnuplot/4.6.5/
- SumatraPDF – PDF/EPS viewer – v.3.1.2
  https://www.sumatrapdfreader.org/
  https://www.sumatrapdfreader.org/download-prev.html

HEPAComp has been used for the graphics in these articles:
https://arxiv.org/abs/1501.07090 1506.08031 1603.03314 1910.11614 2007.03370

HEPAComp website:
https://justmathbg.info/hepacomp.html