Release Notes GAMMA Software, 20231201

Urs Wegmüller, Christophe Magnard, Charles Werner, Andreas Wiesmann
Gamma Remote Sensing AG
Worbstrasse 225, CH-3073 Gümligen
http://www.gamma-rs.ch
1-Dec-2023

Introduction

This information is provided to users of the GAMMA software. It is also available online at https://www.gamma-rs.ch/uploads/media/GAMMA_Software_upgrade_information.pdf.

This release of the Gamma software includes new programs that provide new capability, additional features to existing programs and bug fixes.

Gamma Software on Linux, macOS, and Windows

The Gamma software has been compiled and tested on Linux (different distributions), Apple macOS Ventura (13.6.1), and Windows 10 and 11. Computationally intensive programs such as used in co-registration and resampling and geocoding have been parallelized using the OpenMP API built into the GCC compiler. Processing speed on Linux, macOS, and Windows systems is comparable.

Linux Distribution:

The Gamma software is developed on Ubuntu 22.04 LTS 64-bit Linux and is tested extensively with this distribution. The Gamma software is also available for Ubuntu 20.04 LTS.

Announcement: This is the last time Ubuntu 20.04 LTS will be supported.

Versions of the Software will also be uploaded for RHEL7 based on CentOS7, RHEL8 based on Rocky Linux 8, and RHEL9 based on Rocky Linux 9.

Note for ***CentOS7***: Some Gamma Software programs now require a GDAL version newer than the version provided in CentOS7 standard repository. Hence, newer versions of GDAL and PROJ have to be installed. The installation method is described in the installation instructions (INSTALL_linux.html).

Announcement: Support for RHEL7 / CentOS7 will be available until the mid-2024 upgrade.

Note that Red Hat has ended support for CentOS8 at the end of 2021. Consequently, the Gamma software built for RHEL8 is now built on Rocky Linux 8.

For installation instructions for the binary LINUX distributions see the HTML file INSTALL_linux.html (provided with the distribution E-mail or found in the main directory of the distribution).

Apple MacOS Distribution:

The software in this version has been compiled using macOS Ventura (13.6.1). You will need to install libraries such as GDAL using MacPorts. The build uses GCC 12 compiler. Version are available for Apple Silicon (M1/M2) and Intel x86_64 processors.

For installation instructions for the binary macOS distributions see the HTML file INSTALL_macOS.html (provided with the distribution E-mail or found in the main directory of the distribution).

Windows Distribution:

The Windows version of the Gamma software is compiled with 64-bit support and multi-threaded. The build uses the MINGW64 GCC 13 compiler.

For installation instructions for the binary Windows distributions see the HTML file INSTALL_win64.html (provided with the distribution E-mail or found in the main directory of the distribution).

The Gamma Plugin for ArcGIS is available in all Windows distributions that include the *GEO* or *ISP/DIFF&GEO* modules. Full functionality requires access to the *LAT* module.

On both Windows 10 and 11, it is also possible to install the Windows Subsystem for Linux (WSL2) and run a Linux distribution of the Gamma software on that environment. Instructions for this setup are available in the HTML file INSTALL_wsl.html located in the main directory of the distribution.

Documentation and Program List

The Gamma documentation browser is an HTML based system for viewing the web pages and pdf documents. The documentation browser includes for each module a Contents sidebar on the right side of the screen and a search functionality. The main Gamma documentation browser page *Gamma_documentation.html* is found in the main software directory.

The program *gamma_doc* facilitates the access to the documentation related to a given module or program:

gamma_doc	Opens the main page of the Gamma documentation browser and shows the program list.
gamma_doc DIFF	Opens the DIFF&GEO documentation.
gamma_doc gc_map2	Opens the reference manual web page for *gc_map2*.

Further information related to the GAMMA Software is available online:

General information:
gamma-rs.ch/uploads/media/GAMMA_Software_information.pdf
Technical reports, conference and journal papers:
gamma-rs.ch/uploads/media/GAMMA_Software_references.pdf
Release notes / upgrade information:
gamma-rs.ch/uploads/media/GAMMA_Software_upgrade_information.pdf

In case the program list is incomplete, run the python script program_list.py after successful installation of the Gamma Software in the main folder of the Gamma Software distribution:

```
./program_list.py Gamma_documentation_base.html Gamma_documentation_contents_sidebar.html -a
```

Python and Matlab wrappers

The Gamma Software is integrated into Python and Matlab through wrappers. Gamma Software program calls become Python / Matlab function calls where command line arguments can be used as function arguments, and system outputs can be stored in variables or written to log files. Binary images, point lists and data, parameter files, tab files, can be easily read, inspected, and written using additional functions provided with the wrappers.

The *py_gamma* Python module permits a smooth usage of the Gamma Software within Python scripts as well as within a Python Interactive Development Environment (IDE) such as Spyder or PyCharm or using Jupyter Notebooks. Using *py_gamma*, function arguments can be entered either as positional arguments or as keyword arguments, with the Gamma command line parameter names becoming the keyword names.

In the same way, the Matlab (and Octave) wrapper, composed of *mat_gamma* and *par_file* classes, permits a smooth usage of the Gamma Software within an interactive use of Matlab as well as within Matlab scripts.

Gamma plugin for ArcGIS

The Gamma plugin for ArcGIS permits using some Gamma software functionalities (tools) from ArcGIS Pro (Windows only) using a convenient interface. The Gamma plugin allows to perform the following operations:

- Reading SAR data from various sensors / formats
- Detection, radiometric calibration and geocoding of SAR data
- Co-registration of SLC and MLI SAR images in slant range / azimuth geometry
- Adaptive interferometric coherence estimation
- Multi-temporal processing and filters
- Spatial filtering of 2D SAR images
- Change detection in SAR images

Using ArcGIS ModelBuilder, it is possible to generate dedicated processing chains using the Gamma tools as building blocks. It is also possible to use each Gamma tool as a Python function (ArcGIS Pro is required). Note that the LAT module is required to be able to use all the tools; without the LAT module, only a subset of the tools is available.

Hardware Recommendations

Using multi-core processors (8 or more cores) will bring substantial improvement in processing speed due to parallelization of the code base. There should be at least 8 GB RAM available for each processor core with 16 GB per core recommended. Disk storage requirements for using the Gamma Software effectively depend on the amount of input data and data products that will be produced. Based on our experience we recommend considering at least 16 TB space, especially when working with stacks of Sentinel-1 or very high-resolution data (TerraSAR-X, Cosmo-Skymed) data. The current trend towards larger data products requires substantially increased storage capacities.

GAMMA Software Training Courses

A SAR/INSAR (MSP/ISP/DIFF&GEO/LAT) training at GAMMA (near Bern, Switzerland) is planned for 13. – 16. May 2024. A PSI (IPTA) at GAMMA (near Bern, Switzerland) is planned for 22-Apr-2024 to 25-Apr-2024. See also https://www.gamma-rs.ch/software/training.

Significant Changes in the Gamma Software Modules since the mid 2023 Release

Co-registration programs for Tandem acquisitions

A Tandem acquisition consists of a (normal) monostatic SAR scene (with the reference SAR sensor being the transmitter and the receiver system) and a corresponding bi-static scene (with the reference SAR sensor being the active sensor, the transmitter, and the companion second SAR sensor being the passive receiver-only system).

So far Tandem acquisitions were available from the Shuttle Radar Topographic Mission (SRTM), from TerraSAR/Tandem-X, and from a number of airborne systems. Now satellite Tandem acquisitions are also available from the Chinese L-band constellation LuTan-1 and from the Chines X-band constellation SuperView Neo-2.

To support the processing of Tandem data from the Chinese LT1 and SV2 we updated the coregistration programs *rdc_trans*, *SLC_coreg.py*, and *MLI_coreg.py* with a new option for the Tandem case. In the case of TerraSAR/Tandem-X this option was not needed as the SLC data were provided as already co-registered SLCs.

The phase simulation programs (*phase_sim_orb*, *phase_sim*) already support the Tandem geometry (as this was also needed for TerraSAR/Tandem-X).

Calculate DEM gradient and normal vectors

The program *dem_gradient_vec* was added to calculate the DEM gradient and normal vectors as vectors (in Easting-Northing-Up coordinates). This program complements the program *dem_gradient* that already existed and that calculates the orientation and elevation angles of the DEM gradient and normal vectors. Providing the DEM gradient as a vector has the advantage that vector geometry can be used to project it on the line-of-sight vector (calculated by the program *lk_vec_lt*). The following sequence can be used to project the unit DEM gradient vector onto the unit line-of-sight vector:

lk_vec_lt 20180915.rmli.par CH_A199.dem_par CH_A199.dem CH_A199.20180915.lt_fine - - CH_A199.20180915.lv_ENU

dem_gradient_vec CH_A199.dem_par CH_A199.dem.fspf CH_A199.20180915.grad_ENU 1 1 0

vec_math CH_A199.20180915.grad_ENU CH_A199.20180915.lv_ENU CH_A199.20180915.grad_dot_lv 38993 2

The result of the dot-product (projection) is a value between -1.0 and +1.0. The line-of-sight displacement or displacement rate can be divided by this value to get the corresponding displacement (or displacement rate) value under the assumption that the displacement is along the DEM gradient vector direction. Negative *grad_dot_lv* values correspond to displacements away from the sensor (e.g., landslides on the slopes tilted away from the satellite). Positive *grad_dot_lv* values correspond to displacements towards the sensor (e.g., landslides on the slopes tilted towards the satellite).

Additional model in base_calc

The DIFF&GEO program *base_calc* is used to define the stack structure used for an interferometric time series analysis. In the Dec. 2023 distribution a new *itab_type* was added to generate a stack that includes the pairs as defined with the baseline time-interval and delta-n conditions combined with the pairs of the single reference stack with the indicated reference scenes.

This combination may be a suited compromise between using redundant pairs with short intervals and using a single-reference stack. We used for example the combination of the short interval stack connecting each scene with the subsequent 3 scenes with the single reference stack. Figure 1 shows the related connectivity plots.

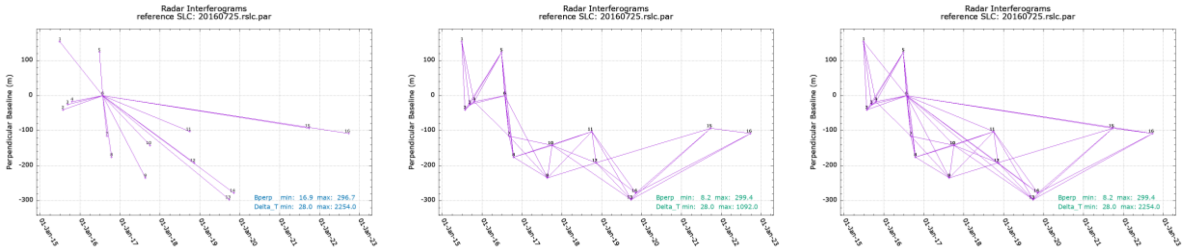


Figure 1. Time series connectivity plots for the single reference stack (left), the multi-reference stack connecting each scene with the subsequent 3 scenes (center) and the new combined stack (right).

New program to calculate coherence based on a pair of co-registered SLC

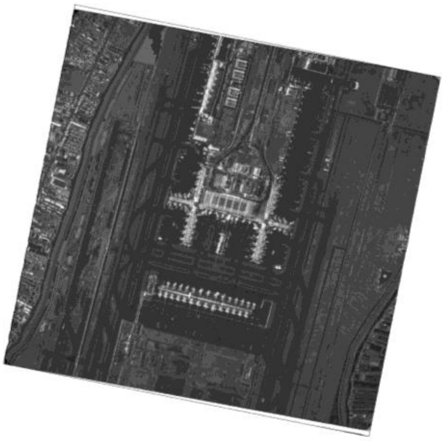
The new LAT program *SLC2cc* supports the calculation of the interferometric coherence based on a pair of co-registered SLCs. Some *GEO/LAT* users are interested in using the coherence in addition to the backscatter for tasks such as landuse classification and change detection.

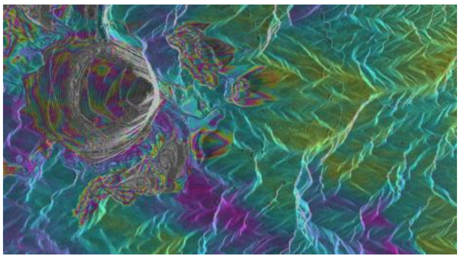
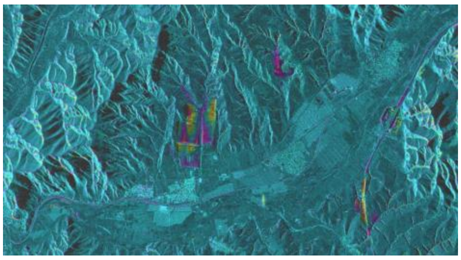
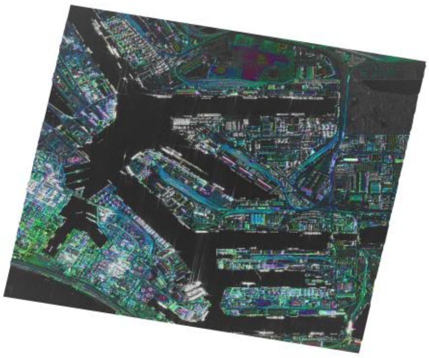
To support this functionality within the *GEO/LAT* modules we added the program *SLC2cc*. As input a pair of co-registered SLCs is used. In the case of Sentinel-1 TOPS data or ScanSAR data it is a pair of mosaicked SLCs.

SLC2cc reads the SLCs and the corresponding multi-look backscatter images (MLIs) and calculates the coherence following a sequence corresponding to *phase_sim_orb*, *SLC_diff_intf* and *cc_ad*. No simulated phases and differential interferograms are written out but only the resulting coherence file.

This functionality can also be called within the Gamma plugin for ArcGIS.

New and updated SAR data readers

New / updated SAR data reader	Short description
par_SICD_SLC supporting now UMBRA SLC data 	The SICD format data reader program now also supports reading UMBRA SLC data processed using Polar Format Algorithm (PFA), as well as data from other sensors processed using PFA. When reading UMBRA SLC data, the PFA data geometry is transformed into the standard 0-Doppler slant-range – azimuth geometry, so that further processing steps can be done using the same programs as for other SLC data in slant-range – azimuth geometry. The image to the left shows a geocoded UMBRA backscatter image (calculated based on an UMBRA SLC in SICD format with PFA geometry).

par_SV2_SLC  Repeat-pass SV2 X-band differential interferogram  Single-pass SV2 X-band Tandem differential interferogram	The program supports reading SLC data of the Chinese X-band Constellation SuperView Neo-2 (LV2). China Siwei is planning to build 28 RS commercial satellites system before 2025, with 16 high resolution (0.2-0.3m resolution) optical satellites, 4 wide-swath (0.7m resolution + 100km swath) optical satellites and 8 SAR (X-band, 0.5m resolution) satellites. In July 2022, the first two X-band SAR satellites for this constellation were successfully launched with the names SuperView Neo-2 01/02 [https://www.chinasiwei.com/zgswen/Company1/SuperViewconstellation/list/SuperViewconstellation.html]. followed by two satellites in 2023. For the implementation of the reader program and the assessment of the data, a spotlight mode scene, a repeat-pass interferometric pair and a Tandem mode interferometric pair were used. The images to the left show small sections of the repeat-pass (top) and Tandem (bottom) differential interferograms. We describe our assessment and the processing done in a short technical report.
par_Capella_SLC  Geocoded Capella Spotlight mode differential interferogram over Antwerp (dt 13 days, Bperp 231m, 23-Jan- 2023 to 5-Feb-2023).	The Capella reader program was updated and a technical report assessing the use of Capella data for interferometry was written.

Gamma Software Demo examples

The access to the Gamma Software Demo examples is limited to Gamma Software users with a valid license. The access information is provided with the software delivery. A list of the Demo examples is available here:

https://gamma-rs.ch/uploads/media/README_Gamma_Software_demo.html.

New / modified demo example:	Contents
Demo for Umbra SLC and GEC data.	A demo example was added to demonstrate the reading of Umbra data. The example chosen is a spotlight mode acquisition over the Bangkok Airport. The Umbra SLC data are in SICD format and are provided in an unusual geometry resulting from the data focusing using the Polar Format Algorithm (PFA). When reading the SLC data, the data geometry is transformed into the standard 0-Doppler slant-range – azimuth geometry, so that further processing steps can be done using the same programs as for other SLC data in slant-range – azimuth geometry. For comparison the GEC product of the same scene is considered.
Demo examples for GAMMA use within ArcGIS Pro.	Demo examples for the use of the GAMMA GEO/LAT tools within ArcGIS Pro are in preparation.

MSP

PALSAR_antpat: Changed name of positional argument from "ant_file" to "antenna" to improve consistency with other programs using antenna files.

ASAR_IM_proc: Changed name of positional argument from "ant_gain" to "antenna" to improve consistency with other programs using antenna files.

ASAR_AP_proc: Changed name of positional argument from "ant_gain1"/"ant_gain2" to "antenna1"/ "antenna2" to improve consistency with other programs using antenna files.

ASAR_proc_all, ERS_proc_all: Changed name of positional argument from "autof_min" to "SNR_min" to improve consistency with other programs.

PALSAR_proc_all, PALSAR_proc_all_WB: Changed name of positional argument from "autof_min" or "autof_snr" to "SNR_min" to improve consistency with other programs.

RSAT_proc_all: Changed name of positional argument from "autof_min" or "autof_snr" to "SNR_min" to improve consistency with other programs.

ISP

bridge_unw: The program "bridge" has been renamed to "bridge_unw" to avoid conflicting with a program named "bridge" used for bridging IP addresses.

par_TX_ScanSAR: Changed name of positional argument from annot_XML to annotation_XML to improve consistency with other programs par_* reader files.

par_SV2_SLC: New program for generating SLC parameter and image files for SuperView Neo-2 SuperView-2 / Gaojing-2 SLC data.

par_SICD_SLC: Now also supports data processed using PFA algorithm, such as SLC data from UMBRA satellites. When reading UMBRA SLC data in polar format the data is transformed into the standard 0-Doppler slant-range – azimuth SLC format. Consequently, the normal programs can then be used for further processing steps as geocoding, co-registration etc.

DIFF&GEO

MLI_coreg.py: New script to run a co-registration sequence with a refinement step (or optionally without) for an MLI to a reference MLI.

phase_sim_orb, phase_sim: Improved accuracy in the calculation of the simulated phase. Default value for [ph_mode] has now been set to "1: subtract phase offset at scene center to improve precision"

dem_gradient_vec: Calculate the gradient vector or the normal surface vector in ENU coordinates. The unit surface gradient vector lies on the surface and points in the direction of the steepest descent. The normal vector is perpendicular to the tangent plane.

coord_to_sarpix_list: The 2 first columns in the output "SAR_coord" list have been swapped and now correspond to the program usage (azimuth_line range_pixel height). Notice that the range/azimuth order had to be flipped for consistency reasons.

Please check if *coord_to_sarpix_list* is used in your scripts and adapt them accordingly.

ScanSAR_coreg_stack.py: The option --2nd_ref was renamed to --sec_ref.

base_calc: New itab type 2 added that includes the multi-reference pairs (optionally with constraints) and all single reference pairs with the specified reference. This is useful for minimizing possible bias in the time-series solution.

data2xyz: Updated to generated double-precision output data values in big-endian. Usage has also been updated for consistency with other program argument names.

dem_xyz: Updated to permit selection of ether single or double precision output. Output is always in big-endian format.

rdc_trans: The program *rdc_trans* has been updated to support both single-pass (Tandem mode) or repeat-pass interferometry. The documentation has been updated to describe the difference of single and repeat-pass interferometry and how to use the new sar_mode option.

SLC_coreg.py: Updated to also support single-pass interferometry. The documentation has been updated to describe the difference of single and repeat-pass interferometry and how to use the new --sar_mode option.

geocoding.py: Now also supports using a constant height value instead of a DEM file.

SLC_diff_intf, SLC_intf: Updated command line usage for parameters rp1_flg and rp2_flg to show the correct values with respect to deskewed (non-zero doppler) and zero-doppler geometry.

data2xyz, data2map: The program data2xyz has been renamed to data2map (and the corresponding html file to data2map.html). The output file is in the map coordinates, not ECEF X,Y,Z, hence the old name was misleading. Updated and improved HTML documentation.

LAT

SLC2cc: New program to estimate the interferometric coherence from SLCs and MLIs using an adaptive estimator. This program allows GEO/LAT users to also consider the interferometric coherence in their signature studies, classification work, change detection etc.

DISP

kml2poly: Improved KML reformatting to better support KMLs with multiple tags on the same

line.

explain_gamma_command.py: New program to provide help information for the positional parameters of gamma programs calls using the terminal.

vis_colormap_bar.py: Added functionality that if max_val is less than min_val, the inversed colormap is used in the colorbar.

visbyte.py, viscpx.py, visdt_pwr.py vismph_pwr.py vispwr.py: Update to use numpy.ndarray.item() to convert 1 element ndarrays to scalar values.

IPTA

There were no relevant modifications to IPTA programs in this period. Nevertheless, the new DIFF&GEO programs *dem_gradient_vec* and *base_calc* are important updates in the context of interferometric time-series analysis. *dem_gradient_vec* calculates the DEM gradient vector (or the normal surface vector) in ENU coordinates. The unit surface gradient vector lies on the surface and points in the direction of the steepest descent. Based on the DEM gradient vector and the look vector (both in ENU coordinates) vector geometry can be used to calculate the sensitivity of the line-of-sight displacement observation to movements in the DEM gradient direction (along the steepest descent). *base_calc* was updated to support a new itab type that includes the multi-reference pairs (optionally with constraints) and all single reference pairs with the specified reference. This is useful for minimizing possible bias in the time-series solution.

TDBP (Time-Domain Back-Projection)

There were no relevant modifications to TDBP programs in this period.

The TDBP programs are available for Ubuntu 22.04 Linux and Windows 11 WSL2 with Ubuntu 22.04 Linux (and legacy 20.04 Ubuntu Linux and Windows 11 WSL2 versions).

The time-domain back-projection focusing program *az_proc_tdbp_gpu* implemented in C/CUDA requires an NVIDIA GPU for parallelized TDBP imaging.

The TDBP module supports image focusing of SAR data from airborne platforms and mobile-mapping platforms such as UAVs and cars, or trains, with high-quality 3-D geometry/motion compensation. Subsequent interferometric/tomographic processing and value-adding is supported. See also our ground motion/slope stability examples acquired with the car-borne Gamma L-band SAR system and processed with the TDBP module: https://www.gamma-rs.ch/L-band-specific-Publications.

Python wrapper

The support for Gamma program calls using keyword parameters has been improved.

Matlab wrapper

-

GIS

With the end-of-2023 distribution the additional module *GIS* is included. All the users with a valid license and running maintenance for the *GEO* or a combination of modules that includes the *ISP/DIFF&GEO* obtain the new additional module *GIS*. Notice that many of the programs of the *GIS* module require access to the *LAT*. Users without the *LAT* will not see the tools requiring *LAT* programs.

The *GIS* programs compose the "***Gamma Plugin for ArcGIS Pro***". This plugin supports running SAR data specific tools in the ArcGIS Pro environment (Windows only). The GIS module consists of several text files (README file, Change log file) and Python programs. The text files are in the main GIS directory, the Python programs and the related documentation (xml files that can be accessed within ArcGIS Pro) are located in the subdirectory scripts.

README_GIS: New text file README_GIS file listing the contents of the GIS module.

CHANGES_GIS: New text file CHANGES_GIS file listing the changes in the GIS module.

arc_gamma.pyt: Main script linking the tools available in the Gamma Plugin for ArcGIS

arc_change_detection.py: Tool to identify change between co-registered SAR images.

arc_coherence.py: Adaptive interferometric coherence estimation from co-registered SLC and MLI images.

arc_coreg.py: Co-register SLC and MLI SAR images in slant range / azimuth geometry.

arc_crop_image.py: Crop SAR images to an area defined within ArcGIS Pro.

arc_gamma_to_GeoTIFF.py: Convert geocoded data from Gamma format to GeoTIFF format and vice-versa.

arc_geocoding.py: Detection, radiometric calibration and geocoding of SAR data.

arc_linear_to_dB.py: Convert geocoded intensity image from linear to dB scale and vice-versa.

arc_multi_look.py: Multi-look SLC, MLI, or geocoded SAR data.

arc_multi_temp.py: Multi-temporal processing and filters.

arc_read_data.py: Read SAR data from various sensors / formats.

arc_spatial_filter.py: Spatial filtering of 2D SAR images using various SAR specific filters.

arc_ScanSAR_mosaic.py: Calculate mosaic of TOPS / ScanSAR SLC burst data.

arc_visualize.py: Visualize images in Gamma format using Gamma display tools.

All packages

SWAP_io: Updated gplot1(), gplot2(), and gplot3d() to use the "noenhance" font option. The default plot size has been increased to 800x600 for gplot1() and gplot2(), and 1024x768 for gplot3d().